

Computational Aspects of Strategic Behaviour in Elections with Top-Truncated Ballots

by

Vijay Menon

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2016

© Vijay Menon 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Understanding when and how computational complexity can be used to protect elections against different manipulative actions has been a highly active research area over the past two decades. Much of this literature, however, makes the assumption that the voters or agents specify a complete preference ordering over the set of candidates. There are many multiagent systems applications, and even real-world elections, where this assumption is not warranted, and this in turn raises a series of questions on the impact of partial voting on the complexity of manipulative actions. In this thesis, we focus on two of these questions. First, we address the question of how hard it is to manipulate elections when the agents specify only top-truncated ballots. Here, in particular, we look at the weighted manipulation problem—both constructive and destructive manipulation—when the voters are allowed to specify top-truncated ballots, and we provide general results for all scoring rules, for elimination versions of all scoring rules, for the plurality with runoff rule, for a family of election systems known as Copeland^α, and for the maximin protocol. Subsequently, we also look at the impact on complexity of manipulation when there is uncertainty about the non-manipulators' votes. The second question we address is the question on what the impact of top-truncated voting is on the complexity of manipulative actions in electorates with structured preference profiles. In particular, we consider electorates that are single-peaked or nearly single-peaked and we show how, for many voting protocols, allowing top-truncated voting reimposes the \mathcal{NP} -hardness shields that normally vanish in such electorates.

Acknowledgements

I am grateful to my advisor, Professor Kate Larson, for her advice and guidance throughout. Kate has been amazingly supportive, has always provided me great feedback and suggestions when I needed it, and has always given me the freedom to pursue my own interests. This work would not have been possible without all of those.

I also thank Professors Shai Ben-David and Eric Blais for agreeing to read and review this thesis, and for offering valuable feedback.

Finally, I would also like to thank my friends here in Waterloo for all the good times and my family for their constant support and love.

Table of Contents

Abstract	iii
Acknowledgements	iv
List of Tables	vii
1 Introduction	1
1.1 Related Work	4
2 Preliminaries	7
2.1 Voting Protocols	8
2.2 Preferences	10
2.2.1 Single-peaked Preferences	10
2.2.2 Nearly Single-peaked Preferences	11
2.3 Manipulative Actions	12
2.4 Computational Complexity	13
3 Complexity of Manipulation in Elections with Top-truncated Ballots	15
3.1 Constructive Manipulation	16
3.1.1 Scoring Rules	16
3.1.2 Scoring Elimination Rules	21
3.1.3 Copeland ^α	23

3.1.4	Maximin	24
3.2	Destructive Manipulation	25
3.3	Impact on Complexity of Manipulation when there is Uncertainty about the Non-manipulators' Votes	28
3.3.1	When only top orders of the non-manipulators are visible	28
3.3.2	When there is only probabilistic information on the non-manipulators	32
3.4	Conclusion	34
4	Reinstating Combinatorial Protections for Manipulation and Bribery in Single-Peaked and Nearly Single-Peaked Electorates	36
4.1	Contributions	37
4.2	Manipulation	38
4.2.1	Single-Peaked Electorates	38
4.2.2	Nearly Single-Peaked Preferences	46
4.3	Bribery	49
4.3.1	Weighted-Bribery in Scoring Rules	49
4.3.2	Weighted-Bribery in Eliminate(veto)	53
4.3.3	Is Weighted-Bribery for Weak-Condorcet Consistent Rules Always Easy?	59
4.4	Is Top-truncated Voting Always Beneficial in SP Electorates?	61
4.5	Conclusion	63
5	Conclusions and Future Work	65
	References	68

List of Tables

3.1	Complexity of CWCM and DWCM with Top-truncated Votes.	35
4.1	Change in complexity of CWCM when moving from complete votes to top-truncated votes in single-peaked electorates.	63
4.2	Change in complexity of CWCM when moving from complete votes to top-truncated votes in k -maverick single-peaked electorates.	64
4.3	Change in complexity of weighted-bribery when moving from complete votes to top-truncated votes in single-peaked electorates.	64

Chapter 1

Introduction

Preference aggregation is an important problem in multiagent settings as there are many scenarios where a group of agents has to make a common decision. The process of arriving at this decision, in turn, has to accommodate the needs and preferences of all the participating agents. A natural, and commonly used, mechanism to achieve this is voting, where all the agents specify their preferences and a previously agreed-upon procedure—called the election system or voting protocol—is used to arrive at the decision. Election problems abound in human contexts. From electing a government to hiring within a CS department to choosing a winner in television reality shows, they are everywhere. Such problems have also been increasingly ubiquitous in multiagent settings. For instance, it has been proposed as a mechanism for web spam reduction [17], for collaborative filtering and recommender systems [41], and for multiagent planning [19].

Given the fact that it is useful and a widely used mechanism, voting, naturally, has been extensively studied and it has been found that it is not without its problems. In fact, the theory of social choice, which focuses on the study of collective decision making processes and procedures, has a number of impossibility results surrounding fundamental issues that arise in running elections. Among many such issues, one natural worry that can arise is “What if the agents try to manipulate the outcome of an election by, say, misreporting their true preferences?”. From the perspective of designing voting systems, this question has been answered through the Gibbard-Satterthwaite Theorem [32, 42] and its generalizations (like the Duggan-Schwartz Theorem [16]) which essentially say that there are no reasonable voting systems which aren’t manipulable. However, despite such results, there still remains the question of how to achieve these manipulations. Starting with the seminal work of Bartholdi, Orlin, Tovey, and Trick through a series of papers [5, 4, 3], there has been much work that has raised this question and has looked into how

computational complexity can be used as a barrier to protect elections against different manipulative actions (see [26, 23] for fairly recent surveys). This thesis too tries to address the question of when computational complexity can be used to protect elections, and in particular we consider the case when the agents are allowed to submit truncated ballots—meaning that they are not obligated to specify their ranking of all the candidates but need to do so only for any subset of them.

Now, the first question that could arise here is: “why truncated ballots?”; after all, there has been a flurry of papers in computational social choice that has studied the complexity of manipulative actions under the assumption that the agents submit complete rankings over the set of candidates. There are a number of reasons why looking at partial preferences in general is important. First, expecting the agents to provide complete ballots is only practical when the number of candidates is small and in practice there are many situations where this might not be the case. As an instance, consider the problem of building a meta-search engine as outlined by Dwork *et al.* [17] by combining search results from individual search engines. In this case, for a given query, if the search results of a particular search engine denotes its “vote”, and the candidates are all the possible search results, then it is easy to see that the votes have to be necessarily truncated before combining. Second, even if the number of candidates were not very large, agents might just be unable to specify a complete ordering because they do not have enough information on them. Third, there are many real-world elections in which voters are allowed to submit partial orderings. For example, apart from the fact that most political elections use the plurality rule where every agent only specifies their most favourite candidate, the elections for the President of Ireland and for the legislative assembly in the Australian Capital Territory uses the Single Transferable Vote where an agent can rank any subset of the candidates. Thus, it is important to understand the repercussions of having partially specified preferences (“partial votes”) in general. While a “partial vote” can refer to any partial ordering over the set of candidates, here we focus on one kind of “partial vote”, namely top-truncated votes. Top-truncated votes are natural in many settings where an agent is certain about its most preferred candidates but is indifferent among the remaining ones or is unsure about them.

In particular, this thesis tries to address the following two questions about elections with truncated ballots.

- While, as outlined above, allowing voters to submit top-truncated ballots can be extremely useful, one of the foremost questions that arises here is: “What happens to the complexity of manipulation if the agents reveal only top-truncated preference orderings?”. It is this question that we try to address first, in Chapter 3. Specifically,

we study the weighted manipulation problem—both constructive and destructive manipulation—when the voters are allowed to specify top-truncated ordering and we provide many general results. Subsequently, we remove the assumption that the manipulators have complete information and instead look at the impact on complexity of manipulation when there is uncertainty about the non-manipulators’ votes.

- Given the set of candidates, one of the most common assumptions made is that they can specify any partial ordering over them. In fact, this is also the assumption based on which the first question, as outlined above, is addressed. However, there are many elections scenarios where the agents have structured preferences—meaning that there are some valid votes that are never cast. It is these type of preferences that we study next. Here, we consider two of the most widely studied such structured preferences, namely single-peaked preferences and nearly single-peaked preferences, and we ask the following question: “What happens to the complexity of manipulative actions when agents have structured preferences and are allowed to submit top-truncated ballots?”. In particular, in Chapter 4, we look at the problem of weighted manipulation and weighted bribery in single-peaked and nearly single-peaked electorates, and we show a number of surprising results regarding the impact of top-truncated ballots on the complexity of these manipulative actions in such electorates.

In trying to address the above mentioned questions, our work attempts to take the line of research which focuses on complexity of manipulative actions in new directions, and improve one existing direction, through the following contributions:

1. We (in Sections 3.1 and 3.2) provide general results for all scoring rules, for elimination versions of all scoring rules, for the plurality with runoff rule, for a family of election systems known as Copeland^α, and for the maximin protocol, thus extending the preliminary work of Narodytska and Walsh on the impact of top-truncated voting on complexity of manipulation [38].
2. We (in Section 3.3), for the first time, look at the impact on complexity of manipulation when there is uncertainty about the non-manipulators’ votes and when the voters are allowed to submit top-truncated ballots.
3. We, for the first time, systematically study the impact of partial voting on manipulative actions in structured preference profiles. In particular, we look at the problem of manipulation and bribery in single-peaked and nearly single-peaked settings when top-truncated ballots are allowed.

4. We (in Sections 4.2 and 4.3) show a number of surprising results where allowing top-truncated voting seems to reinstate combinatorial protections (that normally vanish when voters are only allowed to present complete orders) for many voting rules against manipulation and bribery. In particular, under the assumption that the voters submit complete ballots, we first provide polynomial time algorithms for manipulation and weighted-bribery for certain voting rules in single-peaked and nearly single-peaked settings, thus extending the works of Faliszewski *et al.* for manipulation [27], Brandt *et al.* for bribery [9], and Faliszewski *et al.* for nearly single-peaked electorates [24]. We then show how these polynomial-time problems become \mathcal{NP} -complete when top-truncated ballots are allowed.

The rest of this thesis is organized as follows. In the next section we outline the works that are most related to our results here. Following that, in Chapter 2 we provide the basic definitions regarding elections and the different manipulative action problems considered here. In Chapter 3 we discuss the impact on complexity of manipulation when top-truncated ballots are allowed. Subsequently, in Chapter 4 we move on to scenarios where agents’ preferences have a particular structure and discuss the impact of allowing top-truncated voting on the complexity of manipulation and bribery. Finally, Chapter 5 provides some conclusions and discussions on future work.

1.1 Related Work

The study of computational complexity for two manipulative actions, namely manipulation and control (we do not consider control in this thesis; see, for instance, the paper by Hemaspaandra *et al.* [34] and the references therein), was started through a series of papers by Bartholdi, Orlin, Tovey, and Trick [5, 4, 3]. Subsequently, there has been a flurry of work in this domain of research which has studied the computational complexity of manipulative actions in many different settings and has introduced the study of new manipulative actions—like bribery, as by Faliszewski *et al.* [22]. The reader can refer to some recent surveys for discussions on the evolution of this domain [26, 23].

Broadly, there are two lines of research that are closely related to our work. First is the work on election problems when the agents are allowed to specify partial preferences. Among them, the papers that are most related to our results here are those of Narodytska and Walsh [38], and Fitzsimmons and Hemaspaandra [29]. Narodytska and Walsh were the first to look at complexity of constructive manipulation under top-truncated voting and they provided an analysis for weighted and unweighted elections using three particular

voting protocols, namely Borda, STV, and the Copeland rule [38]. In fact, results in Section 3.1 of this thesis can be considered as a direct extension to their work. The other most relevant work is that of Fitzsimmons and Hemaspaandra where they looked into how the complexity of bribery, control, and manipulation is affected when different types of partial preferences are allowed [29]. While both the above mentioned papers and our results in Chapter 3 assumes the general setting (i.e. one where there is no restriction on the structure of the agents’ preferences), Chapter 4 of this thesis looks at the complexity of manipulation and bribery with top-truncated ballots when the preferences of the agents have some structure.

Additionally, there has also been some work in this line of research that has looked at other election problems when preferences are only partially specified. Among these, the work of Konczak and Lang introduced the possible and necessary winners problem [35], and Xia and Conitzer extended this further to study the possible and necessary winners problem for many different voting rules when the number of candidates are unbounded and the elections unweighted [44]. Furthermore, Lu and Boutilier looked at the multi-winner problem when only partial preferences are provided [37], and Baumeister *et al.* discussed planning various kinds of campaigns in settings where the ballots can be truncated at the top, bottom, or both [6]. The extension-bribery problem they introduced in that paper is closely related to the manipulation problem with top-truncated ballots.

The second line of research that is closely related to results here is the work on structured preference profiles. This line of research has mainly looked at single-peaked preferences and more recently at nearly single-peaked preferences. The notion of single-peaked preferences was introduced by Black [7] and subsequently there has been a lot of work in the social choice literature on the same. Among these, in particular, we note the work of Cantala [10] who introduced the concept of “single-peaked with outside options” which is similar to the notion of single-peaked with top-truncated ballots that we study here in Chapter 4, and the work of Barberá who discussed how properties of different variants of single-peaked preferences change for varying amounts of indifference permitted [1].

In computational social choice, three papers that are in this line of research and are most related to our work are those of Faliszewski *et al.* [27], Brandt *et al.* [9], and Faliszewski *et al.* [24]. The first two discuss manipulation and control, and bribery, respectively, and show how most of the \mathcal{NP} -hardness shields for these manipulative actions vanish in single-peaked settings. The third paper studies the complexity of manipulative actions in nearly single-peaked electorates and shows how in many cases the hardness results evaporate. Our work in Chapter 4, in contrast to all the ones above, follows the theme of reinstating these combinatorial protections by allowing top-truncated voting.

Finally, although the above mentioned papers are by far the most related to our work, it is worth noting that one problem which has received considerable attention from researchers in computational social choice is that of single-peaked consistency where, informally, the task is to determine if a given set of preferences is single-peaked or otherwise (see [2, 15, 21, 20, 36]).

Chapter 2

Preliminaries

The literature on computational social choice typically models an election as a pair $\mathcal{E} = (C, V)$, where $C = \{c_1, \dots, c_m\}$ is the set of candidates and $V = \{v_1, \dots, v_n\}$ is the set of voters and their preferences (also sometimes referred to as the set of votes or ballots). Each voter in turn is modeled as a pair $v_i = (\succ_i, w_i)$, where \succ_i is the preference order of v_i over C and w_i is a positive integer weight associated with them. \succ_i is said to be a complete order (or a complete vote) when it is antisymmetric, transitive, and a total ordering on C . On the other hand, \succ_i is said to be a top-truncated order (or simply, a top order) when it can be a linear order over any non-empty subset of C and where all the unranked candidates are tied and are assumed to be ranked below the ranked candidates. Throughout, a strict preference between two candidates is denoted by “ $>$ ”, while a tie between them is denoted by “ \sim ”.

Example 1. Consider an election scenario with $C = \{c_1, c_2, c_3, c_4\}$. A voter v_i who prefers c_4 the best, c_2 second best, and dislikes c_1 the most has a complete ordering \succ_i which is represented by $(c_4 > c_2 > c_3 > c_1)$, while another voter v_j who likes c_3 the most, c_4 second, but has no opinion on c_1 and c_2 has a top-truncated ordering \succ_j given by $(c_3 > c_4 > c_1 \sim c_2)$.

A preference profile is a vector $P = \langle \succ_1, \dots, \succ_n \rangle$ of individual preferences. An election, \mathcal{E} , is said to be unweighted if for all $v_i \in V$, $w_i = 1$. On the other hand, it is said to be weighted if $\exists v_i \in V$ such that $w_i > 1$.

2.1 Voting Protocols

An election system or a voting protocol takes in a preference profile, P , as input and it outputs a collection $W \subseteq C$ called the winner set. The following are the commonly-studied voting rules that we consider in this thesis. We first present their original definitions which is on complete orders and then discuss how top orders can be handled.

1. **Positional scoring rules:** A positional scoring rule is defined by a scoring vector $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$, where $\alpha_1 \geq \dots \geq \alpha_m$. For each voter $v \in V$, a candidate receives α_i points if it is ranked in the i th position by v . In a scoring rule, all the candidates with highest total score form the winner set. Some examples of scoring rules are the *plurality rule* with $\alpha = \langle 1, \overbrace{0, \dots, 0}^{m-1} \rangle$, the *Borda rule* with $\alpha = \langle m-1, m-2, \dots, 0 \rangle$, and the *veto rule* with $\alpha = \langle 1, \overbrace{\dots, 1}^{m-1}, 0 \rangle$.
2. **Scoring elimination rules:** Let X be any scoring rule. Given a complete ordering, $\text{eliminate}(X)$ is the rule that successively eliminates the candidate with the lowest score according to X . Once a candidate is eliminated, the rule is then repeated with the reduced set of candidates until there is a single candidate left. The elimination order, e , for an elimination rule is the order in which the candidates are successively eliminated. For instance, $e = (c_1, \dots, c_n)$ implies that c_1 was the first candidate to be eliminated while c_n was the last. Here, we mainly consider two scoring elimination rules: *eliminate(Borda)*—which is also known as Baldwin’s rule or Fishburn’s version of Nanson’s rule [40]—and *eliminate(veto)*.
3. **Plurality with runoff:** The plurality with runoff rule proceeds in two steps. In the first step, all the candidates except the top two with the most number of first votes are eliminated. This is followed by a second round where the winner is determined by a pairwise comparison between the top two candidates.
4. **Copeland $^\alpha$:** Let $\alpha \in \mathbb{Q}$, $0 \leq \alpha \leq 1$. In Copeland $^\alpha$ (introduced by Copeland [14] for $\alpha = \frac{1}{2}$ and parameterized by Faliszewski *et al.* [22]), for each pair of candidates, the candidate preferred by the majority receives one point and the other one receives a 0. In case of a tie, both receive α points. Here all the candidates whose total score is at least as high as that of all the other candidates form the winner set.
5. **Maximin:** Let $N_P(c_i, c_j)$ denote the number of voters who prefer c_i over c_j in the preference profile, P . Then the maximin score of c_i is $s_i = \min_{j \neq i} N_P(c_i, c_j)$. The winners in the maximin rule are the ones with the highest score.

To deal with top-truncated orders in positional scoring rules where a voter ranks only k out of the m candidates ($k < m$), we use the following three schemes that were used by Narodytska and Walsh in their preliminary work on manipulation with top orders [38] (Emerson [18] also used the same schemes for the Borda rule).

1. **Round-up:** A candidate ranked in the i th position ($i \leq k$) receives a score of α_i , while all the unranked candidates receive a score of α_m . For any positional scoring rule X , we denote this by X_\uparrow .
2. **Round-down:** A candidate ranked in the i th position ($i \leq k$) receives a score of $\alpha_{m-(k-i)-1}$, while all the unranked candidates receive a score of α_m . For any positional scoring rule X , we denote this by X_\downarrow .
3. **Average score:** A candidate ranked in the i th position ($i \leq k$) receives a score of α_i , while all the unranked candidates receive a score of $\frac{\sum_{k < j \leq m} \alpha_j}{m-k}$. For any positional scoring rule X , we denote this by X_{av} .

Example 2. Let $C = \{c_1, c_2, c_3, c_4\}$ and let v be a voter with top-truncated order \succ_v : ($c_3 > c_1 > c_4 \sim c_2$). Then, in the round-up case, c_3 receives a score of α_1 , c_1 receives a score of α_2 , and both c_2 and c_4 receive α_4 . In the round-down case, c_3 receives a score of α_2 , c_1 receives a score of α_3 , and both c_2 and c_4 receive a score of α_4 , and in the average score case, c_3 receives a score of α_1 , c_1 receives a score of α_2 , and both c_2 and c_4 receive a score of $\frac{\alpha_3 + \alpha_4}{2}$.

In scoring elimination rules and plurality with runoff rule, top-truncated votes are dealt by using the round-up scheme described above. Here, we consider a vote to be valid only until at least one of the candidates listed in it is remaining in the election. In other words, we simply ignore a vote once all the candidates listed in it are eliminated. In the case of Copeland^a and the maximin rule, top orders are dealt by just keeping to the definition which assumes that all the unranked candidates are tied and are ranked below the ranked candidates.

Condorcet-consistent voting rules. An important property of a candidate is that of being a Condorcet winner. A candidate is said to be a Condorcet winner (respectively, a weak-Condorcet winner) if it is preferred over every other candidate by a strict majority of the voters (respectively, by at least half of the voters). While it is not necessary that every election instance has a Condorcet winner or even a weak Condorcet winner, a voting

rule which, on every input that has a weak-Condorcet winner, outputs the set of all weak-Condorcet winners as the set of winners is said to be weak-Condorcet consistent. Among the voting rules defined above, the maximin rule [28] and Copeland¹ [9] are examples of weak-Condorcet consistent rules.

2.2 Preferences

Most work in computational social choice assumes no particular structure for an agent’s preference—i.e. given the set of candidates, C , it is assumed that an agent can specify any linear order over C or a non-empty subset of it (in the case of truncated orders). In fact, this is the assumption that we make for all the results in Chapter 3. However, there are many election scenarios where the preferences of agents have an underlying structure and therefore there are some admissible votes that would never be cast. So, in this context, here, in Chapter 4 of this thesis, we consider two of the most widely studied (see for e.g. [9], [27], [24]) such notions—single-peaked preferences and near single-peaked preferences.

2.2.1 Single-peaked Preferences

The notion of single-peaked preferences, first introduced by Black [7], captures settings where the preferences of a voter are based on a one-dimensional axis. The basic idea here is that every voter has a peak (their most preferred alternative) and that their utility for an alternative decreases the further it is away from this peak. More formally, a preference profile, P , where the preference, \succ_v , of each $v \in V$ is a linear order over C , is said to be single-peaked if there exists a linear order L over C such that for every triple of candidates a, b , and c it holds that

$$(a \ L \ b \ L \ c \vee c \ L \ b \ L \ a) \implies (\forall v \in V) [a \succ_v b \implies b \succ_v c].$$

Although the above definition of single-peakedness assumes that the voters have complete orders, when voters are allowed to present top-truncated ballots, this notion of single-peakedness essentially captures those scenarios where they have a continuous range over L over which their preferences are single-peaked and outside of which they are indifferent among the alternatives. In social choice theory, this notion of single-peaked with top-truncated preferences has been captured as “single-peaked with outside options” in the context of choosing a level of public good by Cantala [10].

Example 3. Let $C = \{c_1, c_2, c_3, c_4\}$ and $c_2 \prec c_4 \prec c_3 \prec c_1$ be a linear order over C . Then, the preference orders $(c_4 \succ c_3 \succ c_2 \succ c_1)$ and $(c_4 \succ c_2 \succ c_3 \succ c_1)$ are both valid complete single-peaked orders, while the preference order $(c_4 \succ c_1 \succ c_2 \succ c_3)$ is not a valid single-peaked order. Also, with respect to the given linear order, $(c_4 \succ c_3 \succ c_2 \sim c_1)$ and $(c_4 \succ c_2 \succ c_3 \sim c_1)$ are both valid top-truncated single-peaked orders, while $(c_4 \succ c_1 \succ c_2 \sim c_3)$ is not.

While the single-peaked model has been extensively studied and has been broadly considered as a useful model for capturing a number of important scenarios like the U.S. presidential elections and elections in committees (see for e.g. [8], [39]), it is not without its demerits. Primarily, it does not capture any electorate that is driven by multidimensional concerns, and even in the case of a unidimensional electorate (one which is focused on a single issue) it is always plausible that there might be a few ‘maverick’ voters who do not conform to single-peakedness. Therefore, although well studied, single-peakedness is often considered an extreme model.

2.2.2 Nearly Single-peaked Preferences

As noted above, although single-peaked preferences are an interesting domain to study, it is often the case that real-world electorates are not truly single-peaked, but are only close to being single-peaked. The notion of “near” single-peakedness was first raised by Conitzer [12] and Escoffier *et al.* [21], and was subsequently systematically studied by Faliszewski *et al.* [24] and Erdélyi *et al.* [20]. Although there are different notions of “nearness” (see [20] for a discussion on the other notions), here we look at only one such notion, namely the maverick notion which is defined below.

Definition 1 (*k*-maverick SP Electorate). *A collection of votes is called a k-maverick SP electorate if all but at most k of the voters are single-peaked consistent with respect to the societal order L.*

When discussing results for manipulative action problems (defined in the next section) in single-peaked and nearly single-peaked electorates, we will follow the model proposed by Walsh [43] and hence assume that the societal order, L , is given as part of the input—or in other words, we assume that the unidimensional issue of concern to an electorate is publicly known (for a more in-depth discussion on why this is a reasonable, and often useful, assumption, see, for example, [9, section 2] and [24, section 4]).

2.3 Manipulative Actions

In this thesis, we consider two weighted manipulative action problems: manipulation and bribery. In particular, we study the Constructive Coalitional Weighted Manipulation (CWCW) problem, the Destructive Coalitional Weighted Manipulation (DWCW) problem, and the Weighted-bribery problem. CWCW and DWCW were first studied by Conitzer *et al.* [13] and are described below. Informally, the main objective in CWCW is to decide if it is possible for a certain set of voters (manipulators) to vote in such a way that it will result in a preferred candidate winning, while in DWCW it is to decide if it is possible for the manipulators to vote in such a way so as to ensure that a particular disliked candidate will not be a winner.

Definition 2 (CWCW). *Given a set of candidates, C , a set of weighted votes, S (preferences of the non-manipulators), the weights for a set of votes, T (manipulators' votes), and a preferred candidate, p , we are asked if there exists a way to cast the votes in T (i.e. for all $v \in T$, determine their \succ_v) so that p is a winner in the election $\mathcal{E} = (C, S \cup T)$.*

Definition 3 (DWCW). *In Destructive Weighted Coalitional Manipulation (DWCW), given a set of weighted votes S (votes of the non-manipulators), the weights for a set of votes T (manipulators' votes), and a disliked candidate, h , we are asked if there exists a way to cast the votes in T so that h does not win the election.*

The complexity-theoretic study of the bribery problem was first introduced by Faliszewski *et al.* [22]. The main objective in the bribery problem is to decide if there exists a subset of voters whose votes can be changed, within a certain budget, to make a preferred candidate win. Here we look at the weighted version of the bribery problem which is described below.

Definition 4 (Weighted-bribery). *Given a set of candidates, C , the set of weighted votes, V , a preferred candidate, p , and a limit, $k \in \mathbb{N}$, we are asked if there exists a way to change the votes of at most k of the voters in V so that it results in p being a winner.*

While there are two possible models—the unique winner model, where the objective is to make the preferred candidate the unique winner, and the non-unique winner model, where the objective is to make the preferred candidate *a* winner—under which the problems defined above can be studied, throughout, unless otherwise specified, we use the non-unique winner model as our standard model.

2.4 Computational Complexity

For most of the \mathcal{NP} -hardness results in this thesis, we use reductions from either the well-known \mathcal{NP} -complete problem Partition (see for e.g. [31]), from a variant of the Partition problem (Partition'), or from a variant of the subset sum problem (Fixed-Difference Subset Sum).

Definition 5 (Partition). *Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ summing to $2K$, we are asked if there exists a subset S_1 of S which sums to K .*

Definition 6 (Partition'). *Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ such that $\sum_i a_i = 2nK$ and $a_i \geq K$, for $1 \leq i \leq n$, we are asked if there exists a subset S_1 such that $\sum S_1 = nK$.*

The \mathcal{NP} -completeness of Partition' can be shown by reduction from the Partition problem.

Lemma 1. *Partition' is \mathcal{NP} -complete.*

Proof. It is easy to see that Partition' is in \mathcal{NP} . To prove \mathcal{NP} -hardness, we show a reduction from an arbitrary instance of the Partition (P1) problem. Let the arbitrary instance be $\{a_1, \dots, a_n\}$ with $\sum_i a_i = 2M$. Now, construct the following instance of Partition'(P2):

1. For each a_i , construct $b_i = a_i + 2M^2$ and $b'_i = 2M^2$.
2. Construct $X_1 = X_2 = 2M^2n + 4M^2 - M$.

Now, it can be seen that $\sum_i (b_i + b'_i) + X_1 + X_2 = 2 \cdot (2n + 2) \cdot 2M^2$ and that each of $b_i, b'_i, X_1, X_2 \geq 2M^2$.

Suppose there exists a partition S_1, S_2 for P1. Then it is easy to see that P2 also has a partition T_1, T_2 such that $\sum T_1 = \sum T_2 = (2n+2)2M^2$, where $T_1 = \{b_i\}_{i|a_i \in S_1} \cup \{b'_i\}_{i|a_i \in S_2} \cup \{X_1\}$ and $T_2 = \{b_i\}_{i|a_i \in S_2} \cup \{b'_i\}_{i|a_i \in S_1} \cup \{X_2\}$.

Conversely, suppose there exists a partition for P2. This implies that there exists two subsets T_1, T_2 such that $\sum T_1 = \sum T_2 = (2n + 2)2M^2$. Now, the first thing to observe is that both X_1, X_2 cannot belong to the same set, because if so then the sum of the rest of the elements $= 4nM^2 + 2M < (2n + 2)2M^2$. So let us assume that $X_1 \in T_1$ and $X_2 \in T_2$. Also, since $X_1 > b_i, \forall i$, we can remove X_1 and X_2 from the two subsets. Removing them

we have $\sum T_1 - X_1 = \sum T_2 - X_2 = M + 2nM^2$. So now we can construct S_1 , and S_2 as follows:

$$\begin{aligned} S_1 &= \{a_i\}_{i|b_i \in T_1} \\ S_2 &= \{a_i\}_{i|b_i \in T_2} \end{aligned}$$

This in turn implies that P1 has a partition. \square

Definition 7 (Fixed-Difference Subset Sum). *Given a set of non-negative integers $S = \{a_i\}_{1 \leq i \leq n}$ summing to $2K$, we are asked if there exists two disjoint subsets S_1, S_2 of S such that $\sum S_1 - \sum S_2 = K$, where $\sum S_i$ denotes the sum of all the elements in the set S_i .*

The \mathcal{NP} -completeness proof of Fixed-Difference Subset Sum is very similar as to the one for Partition'.

Lemma 2. *Fixed-Difference Subset Sum is \mathcal{NP} -complete.*

Proof. It is easy to see that Fixed-Difference Subset Sum is in \mathcal{NP} . To prove \mathcal{NP} -hardness, we show a reduction from an arbitrary instance of the Partition (P1) problem. Let the arbitrary instance be $\{k_1, \dots, k_t\}$ with $\sum_i k_i = 2K$. Now, construct the following instance $\{l_1, \dots, l_t, l'_1, \dots, l'_t\}$ of Fixed-Difference Subset Sum(P2), where $l_i = k_i + 2^{n+i}$, $l'_i = 2^{n+i}$, and $n = \lceil \log 2K \rceil$.

Suppose there exists a partition S_1, S_2 for P1. Then P2 has subsets T_1, T_2 such that $\sum T_1 - \sum T_2 = K + \sum_{i=1}^t 2^{i+n}$, where $T_1 = \{l_i\}_{i|k_i \in S_1} \cup \{l'_i\}_{i|k_i \in S_2}$ and $T_2 = \emptyset$.

Conversely, suppose there exists subsets T_1, T_2 in P2 such that $\sum T_1 - \sum T_2 = K + \sum_{i=1}^t 2^{i+n}$. Now, it is easy to argue that none of l_i or l'_i for $i = 1, \dots, t$ can belong to T_2 , because if so then the second term ($\sum_{i=1}^t 2^{i+n}$) of $\sum T_1 - \sum T_2$ will not be attainable. Therefore, we can construct S_1 and S_2 such that $S_1 = \{k_i\}_{i|l_i \in T_1}$ and $S_2 = \{k_i\}_{i|l_i \notin T_1}$, and this in turn implies that P1 has a partition. \square

Chapter 3

Complexity of Manipulation in Elections with Top-truncated Ballots

A common assumption in much of the research in computational social choice is that the agents fully specify their preferences by providing a complete preference ordering over all the candidates or alternatives. However, there are many practical situations where the agents may not be able to determine a complete ranking over all the candidates or even if they can specify a complete ranking, the voting rule used may not insist that one be provided. Thus, it is important to understand the repercussions of having “partial votes” in general, and in this context one of foremost questions that arises is: “How hard is it to manipulate elections if the agents reveal only partial preference orderings?”. It is this question we try to address in this chapter. In particular, we look at the weighted manipulation problem—both constructive and destructive manipulation—when the voters are allowed to specify one kind of “partial vote”, namely top-truncated votes. Top-truncated votes are natural in many settings where an agent is certain about its most preferred candidates but is indifferent among the remaining ones or is unsure about them.

While there has been some work that has looked at election problems when preferences are only partially specified (see the related work section in Chapter 1), there are three papers in particular that are closely related to the results in this chapter. First is the work by Baumeister *et al.* which discusses planning various kinds of campaigns in settings where the ballots can be truncated at the top, bottom, or both [6]. In this work, they introduced the extension-bribery problem, a special case of which is closely related to the manipulation problem with top-truncated ballots that we consider here (see Section 3.3.1). Second is the work of Narodytska and Walsh where they provide an analysis of constructive manipulation (for both weighted and unweighted voters) for three particular voting protocols, namely

Borda, STV, and the Copeland rule [38]. Finally, there is also a paper by Fitzsimmons and Hemaspaandra which looks into how the complexity of bribery, control, and manipulation is affected when ties are allowed [29]. We note that except for one theorem (Theorem 10), none of our other results overlap with their's as all their results are derived using only one the following protocols: Borda, plurality, t -approval, and Copeland ^{α} .

The rest of this chapter is organized as follows. In Section 3.1 and Section 3.2, we look at broader classes of voting rules and we study constructive and destructive manipulation in weighted elections, respectively. In doing so, we provide general results for the complexity of manipulation for all scoring rules, for elimination versions of all scoring rules, for the plurality with runoff rule, for a family of election systems known as Copeland ^{α} , and for the maximin protocol. Finally, in Section 3.3, we also look at the impact on complexity of manipulation with top-truncated ballots when there is uncertainty about the non-manipulators' votes.

3.1 Constructive Manipulation

In this section, we look at the complexity of constructive manipulation when top-truncated ballots are allowed. We begin by looking at scoring rules and we completely characterize the complexity of manipulation for all 3-candidate scoring rules when using each of the evaluation schemes defined before (see Section 2.1). Note that in the complete votes case, all scoring rules except plurality are known to be \mathcal{NP} -complete for $m \geq 3$ candidates [13, 33].

3.1.1 Scoring Rules

We start with a simple result which shows how computing if a coalition of manipulators can manipulate X_{\uparrow} (for any scoring rule X), $\text{plurality}_{\downarrow}$, plurality_{av} , and veto_{\downarrow} takes only polynomial time.

Theorem 1. *Computing if a coalition of manipulators can manipulate X_{\uparrow} , veto_{\downarrow} , $\text{plurality}_{\downarrow}$, or plurality_{av} , where X is any positional scoring rule, with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof. For the X_{\uparrow} , veto_{\downarrow} , and plurality_{av} , the manipulators can simply check if all of them voting for p alone will make it a winner. If not, they cannot make p a winner.

In case of plurality $_{\downarrow}$, they can check if all of them placing p at the top and all the other candidates in arbitrary order can make p a winner. \square

Next, we look at complexity of manipulation for any scoring rule that is not isomorphic to plurality or veto (we say that a scoring rule is isomorphic to another if the scoring vector of one is a linear transformation of the other) and when using the round-down evaluation scheme.

Theorem 2. *For any 3-candidate positional scoring protocol X that is not isomorphic to plurality or veto, CWCM with top-truncated votes in X_{\downarrow} is \mathcal{NP} -complete.*

Proof. Since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 > \alpha_3 = 0$ (because $\alpha_1 = \alpha_2$ is isomorphic to veto, $\alpha_2 = \alpha_3$ is isomorphic to plurality, and α_3 can be taken to be zero since translating the scores in a scoring rule does not affect the outcome of the rule). Also, note that if the three candidates are p, a , and b , each manipulator votes in one of the following ways: $(p > a \sim b), (p > a > b), (p > b > a)$, where for $(p > a \sim b)$ candidate p gets a score α_2 .

The problem is in \mathcal{NP} since winner determination for any scoring rule can be done in polynomial time. To show \mathcal{NP} -hardness, we proceed by considering three cases: 1) $\alpha_1 > \frac{3}{2}\alpha_2$ 2) $\alpha_1 < \frac{3}{2}\alpha_2$ 3) $\alpha_1 = \frac{3}{2}\alpha_2$. For the first two cases, we reduce an arbitrary instance of the Partition problem to an instance of CWCM, and for the third case we show a reduction from the Fixed-Difference Subset Sum problem.

Case 1: $\alpha_1 > \frac{3}{2}\alpha_2$. Given a Partition instance $\{k_i\}_{1 \leq i \leq t}$ summing to $2K$, construct the following instance of CWCM, where p, a , and b are the three candidates. In S , let there be two voters, each of weight $(2\alpha_1 - \alpha_2)K$, voting $(a > b > p)$ and $(b > a > p)$, respectively. As a result, a and b have a score of $(2\alpha_1 - \alpha_2)(\alpha_1 + \alpha_2)K$ each. In T , let each k_i have a vote of weight $(\alpha_1 + \alpha_2)k_i$.

Suppose there exists a partition. Let those manipulators in one partition vote $(p > a > b)$ and those in the other vote $(p > b > a)$. Then the score of p, a and b , is $2\alpha_1(\alpha_1 + \alpha_2)K$, and so p is a winner (since all of them have the same score and we are considering the non-unique winner model).

Conversely, suppose there exists a manipulation in favour of p . Let x, y , and z be the sum of the k_i 's of the manipulators in T who vote $(p > a > b), (p > b > a)$, and $(p > a \sim b)$, respectively. So now, the score of p is $((x + y)\alpha_1 + z\alpha_2)(\alpha_1 + \alpha_2)$, while that of a and b is $((2\alpha_1 - \alpha_2)K + x\alpha_2)(\alpha_1 + \alpha_2)$ and $((2\alpha_1 - \alpha_2)K + y\alpha_2)(\alpha_1 + \alpha_2)$, respectively. Since there

exists a successful manipulation, score of p should be at least as large as that of a , and so we have $((x+y)\alpha_1 + z\alpha_2)(\alpha_1 + \alpha_2) \geq ((2\alpha_1 - \alpha_2)K + x\alpha_2)(\alpha_1 + \alpha_2)$. Using the fact that $x + y + z = 2K$, this simplifies to

$$(K - x)\alpha_2 \geq z(\alpha_1 - \alpha_2). \quad (3.1)$$

Again, the score of p should be at least as large as that of b , so we have

$$(K - y)\alpha_2 \geq z(\alpha_1 - \alpha_2). \quad (3.2)$$

Adding (3.1) and (3.2) and simplifying it, we have $z(2\alpha_1 - 3\alpha_2) \leq 0$. Now, since we assumed $\alpha_1 > \frac{3}{2}\alpha_2$, this implies that $z \leq 0$. But z cannot be less than 0, so it has to be equal to 0. Plugging $z = 0$ in (3.1) and (3.2), we have $x \leq K$ and $y \leq K$ respectively. This together with the fact that $x + y + z = 2K$ implies that $x = y = K$, and therefore there exists a partition.

Case 2: $\alpha_1 < \frac{3}{2}\alpha_2$. Given a Partition instance, construct the following instance of CWCM. In S , let there be a voter of weight $15K$ voting $(b > a > p)$, a voter of weight $5K$ voting $(b > p > a)$, a voter of weight $11K$ voting $(a > p > b)$, a voter of weight $9K$ voting $(a > b > p)$, and a voter of weight $7K$ voting $(p > b > a)$ and $(p > a > b)$ each. As a result, the scores of a , b , and p are $(20\alpha_1 + 22\alpha_2)K$, $(20\alpha_1 + 16\alpha_2)K$, and $(14\alpha_1 + 16\alpha_2)K$ respectively. In T , let each k_i have a vote of weight $6k_i$.

Suppose there exists a partition. Let those manipulators in one partition (who weight to $6K$) vote $(p > b > a)$ and those in the other vote $(p > a \sim b)$. Then the score of all the three candidates is $20\alpha_1 K + 22\alpha_2 K$, and so p is a winner.

Conversely, suppose there exists a manipulation in favour of p . Let x, y , and z be the total weight of manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$ respectively. So now, the score of p is $(x+y)\alpha_1 + z\alpha_2 + (14\alpha_1 + 16\alpha_2)K$, while that of a and b is $(20\alpha_1 + 22\alpha_2)K + x\alpha_2$ and $(20\alpha_1 + 16\alpha_2)K + y\alpha_2$ respectively. Since there exists a successful manipulation, the score of p should be at least as large as that of a , and so we have $(x+y)\alpha_1 + z\alpha_2 + 14\alpha_1 K + 16\alpha_2 K \geq (20\alpha_1 + 22\alpha_2)K + x\alpha_2$. Using the fact that $x + y + z = 12K$, this simplifies to

$$6(\alpha_1 - \alpha_2)K - x\alpha_2 \geq z(\alpha_1 - \alpha_2). \quad (3.3)$$

Again, the score of p should be at least as large as that of b , so we have

$$6\alpha_1 K - y\alpha_2 \geq z(\alpha_1 - \alpha_2). \quad (3.4)$$

Adding (3.3) and (3.4) and simplifying it, we have $(6K - z)(2\alpha_1 - 3\alpha_2) \geq 0$. Now, since we assumed $\alpha_1 < \frac{3}{2}\alpha_2$, this implies that $(6K - z) \leq 0$, or $z \geq 6K$. Plugging $z \geq 6K$ in (3.3) and (3.4), we have $x \leq 0$, and $y \leq 6K$, respectively. But then x cannot be less than 0, so it has to be equal to 0, and this in turn results in $z \leq 6K$ in (3.3). But again, z cannot not be both greater than and lesser than equal to $6K$. So, z has to be equal to $6K$, and since $x + y + z = 12K$, $y = 6K$. This implies there exists a partition.

Case 3: $\alpha_1 = \frac{3}{2}\alpha_2$. Consider the same instance of CWCM as in case 2. The scores of a , b , and p are $(20\alpha_1 + 22\alpha_2)K$, $(20\alpha_1 + 16\alpha_2)K$, and $(14\alpha_1 + 16\alpha_2)K$ respectively. In T , let each k_i have a vote of weight $6k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. Let those manipulators who are in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let all the remaining manipulators vote (p) . If x, y , and z denote the sum of k_i 's of the manipulators who vote for $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of p, a , and b are $9(x + y)\alpha_2 + 6z\alpha_2 + 37\alpha_2K$, $52\alpha_2K + 6x\alpha_2$, and $46\alpha_2K + 6y\alpha_2$, respectively. Now if there existed a manipulation, then the score of p has to be at least as large as that of a and b . Let us consider p and a first. Whatever follows can be replicated for b . Suppose $score(p) \geq score(a)$. This implies $9(x + y)\alpha_2 + 6z\alpha_2 + 37\alpha_2K \geq 52\alpha_2K + 6x\alpha_2$. Simplifying this we have, $x + 3y + 2z \geq 5K$. But since $y - x = K$ and $x + y + z = 2K$, we know that $x + 3y + 2z = 5K$, and hence our assumption that $score(p) \geq score(a)$ is true. Doing the same with respect to p and b , we will see that $score(p) = score(b)$. As a result, we can conclude that existence of S_1, S_2 such that $\sum S_1 - \sum S_2 = K$ results in a successful manipulation for p .

Conversely, suppose there exists a manipulation in favour of p . This implies that the score of p is at least as much as that of a , and from above we know that this in turn results in the inequality $x + 3y + 2z \geq 5K$. Using the fact that $x + y + z = 2K$, we have $y - x \geq K$. Similarly, comparing p and b we have, $y - x \leq K$. But then, $y - x$ cannot be both greater and lesser than equal to K at the same time. So $y - x$ has to be equal to K , and this in turn implies that there exists two sets S_1, S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$. \square

Our final result for scoring rules is when the evaluation scheme used is average score.

Theorem 3. *For any 3-candidate positional scoring protocol X that is not isomorphic to plurality, CWCM with top-truncated votes in X_{av} is \mathcal{NP} -complete.*

Proof. Like in Theorem 2, since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 \geq \alpha_2 > \alpha_3 = 0$.

Also, note that if the three candidates are p, a , and b , each manipulator votes in one of the following ways: $(p > a \sim b)$, $(p > a > b)$, $(p > b > a)$, where for $(p > a \sim b)$ candidate p gets a score α_1 , a and b receive a score of $(\alpha_2/2)$.

The proof uses a reduction from the Fixed-Difference Subset Sum problem and is very similar to the one in case 3 of Theorem 2. Given an instance of Partition, construct the following instance of CWCM where in S we have a voter of weight $(4\alpha_1 + \alpha_2)K$ voting $(b > a > p)$, a voter of weight $(2\alpha_1 - \alpha_2)K$ voting $(a > b > p)$, and a voter of weight $2(\alpha_1 + \alpha_2)K$ voting $(a > p > b)$. As a result, the scores of a, b , and p are $(4\alpha_1 + \alpha_2)(\alpha_1 + \alpha_2)K$, $(4\alpha_1 - \alpha_2)(\alpha_1 + \alpha_2)K$, and $2\alpha_2(\alpha_1 + \alpha_2)K$, respectively. In T , let each k_i have a vote of weight $2(\alpha_1 + \alpha_2)k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. Let those manipulators who are in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let all those remaining vote $(p > a \sim b)$. If x, y , and z denote the sum of the k_i 's of the manipulators who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of p, a , and b are $(4\alpha_1 + 2\alpha_2)(\alpha_1 + \alpha_2)K$, $((4\alpha_1 + \alpha_2)K + 2(x\alpha_2 + z\alpha_2/2))(\alpha_1 + \alpha_2)$, and $((4\alpha_1 - \alpha_2)K + 2(y\alpha_2 + z\alpha_2/2))(\alpha_1 + \alpha_2)$, respectively. Now if there existed a manipulation, then the score of p has to be at least as large as that of a and b . Let us consider p and a first. Whatever follows can be replicated for b . Suppose $score(p) \geq score(a)$. This implies that $(4\alpha_1 + 2\alpha_2)K \geq (4\alpha_1 + \alpha_2)K + 2x\alpha_2 + z\alpha_2$. Simplifying this we have, $2x + z \leq K$. But since $y - x = K$ and $x + y + z = 2K$, we know that $2x + z = K$, and hence our assumption that $score(p) \geq score(a)$ is true. Doing the same with respect to p and b , we will see that $score(p) = score(b)$. As a result, we can conclude that existence of S_1, S_2 such that $\sum S_1 - \sum S_2 = K$ results in a successful manipulation for p .

Conversely, suppose there exists a manipulation in favour of p . This implies that the score of p is at least as much as that of a , and from above we know that this in turn results in the inequality

$$2x + z \leq K. \quad (3.5)$$

Similarly, comparing p and b we have,

$$2y + z \leq 3K. \quad (3.6)$$

Now, using the fact that $x + y + z = 2K$, we know that inequality (3.5) reduces to $y - x \geq K$ and inequality (3.6) reduces to $y - x \leq K$. But then, $y - x$ cannot be both greater and lesser than equal to K at the same time. So $y - x$ has to be equal to K , and this in turn implies that there exists two sets S_1, S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$. \square

3.1.2 Scoring Elimination Rules

We now consider scoring elimination rules and first look at how top-truncated voting affects the complexity of manipulation in $\text{eliminate}(\text{veto})$. Following that we prove a general result for all scoring elimination rules.

Theorem 4. *For $\text{eliminate}(\text{veto})$, in the unique winner model, any manipulation that can be achieved by casting top-truncated votes can be achieved if only complete votes were allowed.*

Proof. Consider an arbitrary set W of top-truncated votes which—along with the set S of non-manipulators’ votes—results in an elimination order $e = (c_1, c_2, \dots, c_m = p)$, where p is the preferred candidate, and c_i is the candidate eliminated in the i th round. Now, consider the set of votes X such that each vote in W is replaced by $(p = c_m > c_{m-1} > \dots > c_1)$. X along with S results in the same elimination order e . Therefore, we see that any manipulation that can be achieved by a set of top-truncated votes can be achieved by casting an equivalent set of complete votes. \square

Since top-truncated voting does not encourage more strategic voting, it follows that for bounded number of candidates—i.e. when there are only a (small) fixed number of candidates—we can use the result by Coleman and Teague who showed that CWCM for $\text{eliminate}(\text{veto})$ is in \mathcal{P} when the votes are complete [11].

Corollary 5. *In the unique winner model, computing if a coalition of manipulators can manipulate $\text{eliminate}(\text{veto})$ with weighted top-truncated votes takes polynomial time for bounded number of candidates.*

Next, we consider elimination versions of 3-candidate scoring rules in general and show that CWCM with top-truncated votes is \mathcal{NP} -complete for elimination version of any scoring rule that is not isomorphic to veto. For this, we first show that top-truncated voting does not change the complexity of a related problem, Anti-WCM, for any scoring rule.

Definition 8 (Anti-WCM). *Given a set, S , of weighted votes, the weights for a set of votes, T , and a disliked candidate, d , we are asked if there exists a way to cast the votes in T so that it results in d receiving the lowest score.*

Theorem 6. *Top-truncated voting does not change the worst-case complexity of Anti-WCM for any scoring protocol.*

Proof. Assume there exists an arbitrary set W of top-truncated votes which results in d receiving the lowest score. For each of the top-truncated votes in W , let us complete them in the following way: If d is included in the vote, append all the other candidates who are not part of it in any arbitrary order. If d is not there, place d at the bottom of the preference ordering (as the m th preferred candidate) and the rest in any arbitrary order. Completing the votes as above does not change the candidate with the lowest score. \square

As a result of the above theorem, we have the following corollary which says that for any scoring rule not isomorphic to veto, Anti-WCM with top-truncated votes is \mathcal{NP} -complete. Note that the corollary here is based on the result of Coleman and Teague who proved that Anti-WCM is \mathcal{NP} -complete for all scoring rules not isomorphic to veto [11, Corollary 10.1].

Corollary 7. *For any scoring rule with $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$, Anti-WCM with top-truncated votes is in \mathcal{P} if $\alpha_1 = \dots = \alpha_{m-1}$ and is \mathcal{NP} -complete otherwise.*

Next, we use the above result to show our main result concerning all scoring elimination rules except eliminate(veto).

Theorem 8. *For any 3-candidate scoring rule X that is not isomorphic to veto, CWCM with top-truncated votes in $\text{eliminate}(X)$ is \mathcal{NP} -complete.*

Proof. Since there are only three candidates and the corresponding positional scoring rule is not isomorphic to veto, the scoring vector for the same is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 \geq \alpha_3 = 0$. Showing that the problem is in \mathcal{NP} is easy. To show \mathcal{NP} -hardness, we use a reduction from an arbitrary instance of Anti-WCM $\langle S, T, h \rangle$ with 3 candidates, where a, b , and h are the three candidates, and h is the disliked candidate. In the CWCM instance we construct, we use the same set of candidates, the same set of manipulators T , and to the S from the Anti-WCM instance we add the following set S' of voters such that K is greater than the sum of the weights in S and T combined. In each case, we add 1 voter of the corresponding type and weight specified below.

$$K : (a > b > h)$$

$$K : (b > h > a)$$

$$K : (h > a > b)$$

We set a to be the preferred candidate in the CWCM instance.

Suppose there was a way to make h receive the lowest score in the Anti-WCM instance. This implies that h receives the lowest score in the CWCM instance since all the three

candidates a, b , and h are tied in S' (each of them receive a score of $\alpha_1 K + \alpha_2 K$ from S'). Therefore, h will be eliminated in the first round, and following that the votes in S' would be:

$$K : (a > b)$$

$$K : (b > a)$$

$$K : (a > b)$$

And now since there are only two candidates, for any scoring rule X , $\text{eliminate}(X)$ is equivalent to $\text{eliminate}(\text{veto})$. Hence, the candidate with the most number of last preferences will be eliminated next. In our case this is b since it has K extra last preferences. Therefore, a wins, thus ensuring a successful manipulation in the CWCM instance.

Conversely, suppose there exists a successful constructive manipulation for a in the CWCM instance. Now, this is possible only if h is eliminated in the first round, because otherwise if b is eliminated in the first round, then a will have K extra last preferences thus resulting in its elimination in the second round. This in turn implies that there is a successful manipulation against h in the Anti-WCM instance. \square

Since the plurality with runoff rule is the same as STV when there are only three candidates, we have the following corollary.

Corollary 9. *For the 3-candidate plurality with runoff rule, CWCM with top-truncated votes is \mathcal{NP} -complete.*

3.1.3 Copeland $^\alpha$

Narodytska and Walsh showed that CWCM with top-truncated votes in the Copeland rule (Copeland $^{0.5}$) is \mathcal{NP} -complete for four candidates [38]. Additionally, they also conjectured that the result holds when the number of candidates is three. Here we prove that conjecture, and also show that our hardness result holds for all rational $\alpha \in [0, 1)$. We note that the following result has also been independently obtained by Fitzsimmons and Hemaspaandra [29].

Theorem 10. *Let α be a rational number with $0 \leq \alpha < 1$. For Copeland $^\alpha$, CWCM with top-truncated votes is \mathcal{NP} -complete for three candidates.*

Proof. It is easy to show that the problem is in \mathcal{NP} . To show that it is \mathcal{NP} -hard, we use a reduction from an arbitrary instance of Fixed-Difference Subset Sum problem. Let p, a ,

and b be the three candidates. In S , let there be a voter of weight $3K$ voting $(a > b > p)$ and a voter of weight K voting $(b > a > p)$. In T , let each k_i have a vote of weight $2k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. In Copeland^a, it can be assumed that all the manipulators rank p first. So, let the manipulators in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let the rest vote for $(p > a \sim b)$. If $N_V(r, s)$ denotes the total number of votes in V which rank r prior to s and $D_V(r, s) = N_V(r, s) - N_V(s, r)$, then $D_{S \cup T}(p, a) = 0$ and $D_{S \cup T}(p, b) = 0$. Therefore, the score of p , $score(p) = 2\alpha$. Also since $\sum S_1 - \sum S_2 = K$, $D_T(a, b) = -2K$, while $D_S(a, b) = 2K$. Therefore, $D_{S \cup T}(a, b) = 0$ and so, both receive a score 2α . Since all of them have the same score, p is a winner.

Conversely, suppose there exists a successful manipulation in favour of p . If x, y , and z , denote the sum of k_i 's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then without taking into account the pairwise election between a and b in T , the score of p , a , and b is $2\alpha, 1 + \alpha$, and α , respectively. Now since $2\alpha < 1 + \alpha$ for all rational $\alpha \in [0, 1)$, therefore, the only way p would win this is if including the pairwise election between a and b in T results in a tie between them. So this implies that $D_{S \cup T}(a, b) = 2K + 2x - 2y = 0$ and that $y - x = K$. This in turn implies that there exists sets S_1 and S_2 such that $\sum S_1 - \sum S_2 = K$, where $y = \sum S_1$ and $x = \sum S_2$. \square

3.1.4 Maximin

Although we have seen instances like in scoring rules with the round-up evaluation scheme (Theorem 1) where top-truncated voting decreases the complexity of manipulation (as compared to the complete votes case), they aren't really conclusive in the sense that one could question the choice of the evaluation scheme that in turn caused the result. Clearly, the round-up evaluation scheme isn't a good choice as what we're essentially doing by employing the same is to encourage the manipulators to treat it just as a plurality-type rule. Therefore, one question that could arise is: "Is there a voting system for which no matter how the top-truncated votes are dealt with the complexity of manipulation with top-truncated ballots decreases?". We answer this question below in the affirmative. We see that as long as all the unranked candidates are considered tied and are assumed to be ranked below the ranked candidates (which is the natural definition of a top-truncated vote), the complexity of manipulation with top-truncated ballots is in \mathcal{P} for the maximin rule. Note that CWCW for the maximin rule is known to be \mathcal{NP} -complete for four candidates when we consider only complete votes [13, Theorem 8]¹.

¹Although Conitzer *et al.* uses the unique winner model [13], it can be verified that the result holds for non-unique winner model as well.

Theorem 11. *Computing if a coalition of manipulators can manipulate the maximin protocol with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof. Let W be an arbitrary set of top-truncated votes which—along with the set S of non-manipulators’ votes—results in p being a winner. Since in the maximin rule moving p to the top will never hurt p , we can safely assume that all the votes in W have p at the top. Now for every vote in W , replace it by $(p > c_1 \sim \dots \sim c_{m-1})$, where c_1, \dots, c_{m-1} are the other candidates in an m -candidate election. By doing so we see that p ’s score does not change. Also, note that for all the other candidates their scores can only decrease or stay the same, but can never increase. Therefore, any constructive manipulation achieved for p can be achieved if all manipulators just vote $(p > c_1 \sim \dots \sim c_{m-1})$. \square

3.2 Destructive Manipulation

In this section, we look at the complexity of destructive manipulation when top-truncated ballots are allowed. We begin by looking at a broad class of rules for which top-truncated voting has no impact on strategic voting. This class consists of all voting rules where the candidates are assigned numerical scores based on the votes and are monotone, meaning that if a voter v_i changes his or her vote (from \succ to \succ') in such a way that $\{b : a > b\} \subseteq \{b : a >' b\}$, then a ’s score will not decrease (or informally, more support for a candidate will not decrease its score). Note that although elimination versions of scoring rules like STV and the Baldwin’s rule are based on numerical scores, they are not monotone, and hence aren’t part of the class of voting rules we consider in the theorem below.

Theorem 12. *For any voting rule that is monotone and is based on numerical scores, any destructive manipulation that can be achieved by casting top-truncated votes can be achieved if only complete votes were allowed.*

Proof. Consider a voting rule X that is monotone and is based on numerical scores. Let the destructive manipulation be against the candidate h . Now, suppose there exists an arbitrary set of top-truncated votes W that—along with the set S of non-manipulators’ votes—results in the destructive manipulation of h in X . Since X is based on scores we will have a final ordering of the candidates after the election. Let $e : c_1 (\neq h) > c_2 > \dots > c_m$ denote that ordering. Next, consider the set of votes W' which is formed by completing the votes in W in the following way: replace each vote in W by placing c_1 at the top, h at the bottom (i.e. at the m th position), and the rest of the candidates in any arbitrary order. Since X is monotone, W' along with S cannot result in the score of c_1 decreasing and

nor can it result in the score of h increasing. Therefore, if W resulted in the destructive manipulation of h , then so should W' . \square

Since top-truncated voting has no impact on destructive manipulation in rules that are monotone and are based on numerical scores, it follows that for bounded number of candidates we can use the result by Conitzer *et al.* who showed that DWCM was in \mathcal{P} for all of them when only complete votes are allowed [13].

Corollary 13. *DWCM with top-truncated votes is in \mathcal{P} for all scoring rules, for the maximin rule, and for Copeland ^{α} .*

Next, we consider the elimination versions of all 3-candidate scoring rules and we show how for all scoring rules that are not isomorphic to veto DWCM is \mathcal{NP} -complete. We prove this by using a similar reduction as in Theorem 8 from an arbitrary instance of Anti-WCM.

Theorem 14. *For any 3-candidate positional scoring rule X that is not isomorphic to veto, DWCM in $\text{eliminate}(X)$ with top-truncated votes is \mathcal{NP} -complete.*

Proof. Since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 \geq \alpha_3 = 0$. Showing that the problem is in \mathcal{NP} is easy. To show \mathcal{NP} -hardness, we use a reduction from an arbitrary instance of Anti-WCM $\langle S, T, h \rangle$ with 3 candidates (see Corollary 7), where a, b , and h are the three candidates, and h is the disliked candidate. In the DWCM instance we construct, we use the same set of candidates, the same set of manipulators T , and to the S from the Anti-WCM instance we add the following set S' of voters such that K is greater than the sum of the weights in S and T combined. In each case, we add 1 voter of the corresponding type and weight specified below.

$$\begin{aligned} K &: (a > h > b) \\ 2K &: (h > a > b) \\ K &: (b > h > a) \\ 2K &: (h > b > a) \\ 3K &: (a > b \sim h) \\ 3K &: (b > a \sim h) \end{aligned}$$

We set h to be the disliked candidate in the DWCM instance.

Suppose there was a way to make h receive the lowest score in the Anti-WCM instance. If $\text{score}_S(a)$ denotes the score candidate a receives from S , then this implies that $\text{score}_{S \cup T}(a)$

$> \text{score}_{S \cup T}(h)$, and $\text{score}_{S \cup T}(b) > \text{score}_{S \cup T}(h)$. Also, note that all the three candidates a, b , and h are tied in S' as each of them receive a score of $4\alpha_1 K + 2\alpha_2 K$. This in turn implies that h receives the lowest score in the DWCM instance, and so will be eliminated in the first round. Thus, existence of a successful manipulation in the Anti-WCM instance ensures the existence of a successful manipulation in the DWCM instance.

Conversely, suppose there exists a successful destructive manipulation against h in the DWCM instance. We first show that this is possible only if h is eliminated in the first round in $\text{eliminate}(X)$. To do so, let us assume it were not the case and that one of a or b was eliminated in the first round. Let us consider a first. If a was eliminated in the first round then the votes in S' would now be:

$$\begin{aligned} K &: (h > b) \\ 2K &: (h > b) \\ K &: (b > h) \\ 2K &: (h > b) \\ 3K &: (b > h) \end{aligned}$$

Since the elimination of a means that there are only two candidates remaining, from now on we can assume our protocol to be equivalent to plurality (since any scoring rule is equivalent to plurality when there are only two candidates). So now, $\text{score}_{S'}(h) - \text{score}_{S'}(b) = K$ and since K is greater than sum of the weights in S and T combined this implies that in the subsequent round b will be eliminated, thus resulting in h winning the DWCM instance. Therefore, there cannot be a destructive manipulation against h in the DWCM instance if a is eliminated in the first round. Similarly, by doing things identically for candidate b , we can see that a destructive manipulation against h will not be possible if b is eliminated in the first round. This in turn leads us to conclude that a successful destructive manipulation against h is possible only if h is eliminated in the first round. But then, since all the three candidates are tied in S' , the only way this can happen is if h receives the lowest score in S . Or in other words, a successful destructive manipulation against h in the DWCM instance is possible only if there exists a successful manipulation against h in the Anti-WCM instance. \square

Again, since the plurality with runoff rule is equivalent to STV when there are only three candidates, we have the following corollary.

Corollary 15. *For the 3-candidate plurality with runoff rule, DWCM with top-truncated votes is \mathcal{NP} -complete.*

3.3 Impact on Complexity when there is Uncertainty about the Non-manipulators' Votes

So far, we have looked at the complexity of manipulation with top-truncated ballots when the manipulators have complete information on the non-manipulators' votes. Although this is a useful setting to study, given that it enables us to look at the hardness of manipulation without having to worry about the complexities that are introduced as part of the uncertainty model, the assumption that the manipulators will have complete information isn't always realistic. Therefore, we now look at how incomplete information about the non-manipulators impacts the complexity of manipulation with top-truncated ballots.

To model the incomplete information setting, we consider the following two scenarios and we study each of them separately. Note that the second setting has been studied by Conitzer *et al.* for the case of complete votes [13].

1. What if only top-truncated preference orderings of the non-manipulators were visible to the manipulators?
2. What if the manipulators have only probabilistic information on the votes of the non-manipulators?

3.3.1 When only top orders of the non-manipulators are visible

Before we look at the problem of manipulation, let us introduce two other problems: i) the problem of evaluating a candidate's winning probability when there's uncertainty about the votes and ii) the weighted version of the extension-bribery problem which in turn was introduced by Baumeister *et al.* [6].

Definition 9 (Evaluation under Top-truncated Uncertainty). *We are given a set, S , which is a partially-revealed top-truncated set of votes of a certain set, S' (which in turn may contain complete or top-truncated ballots themselves), the weights of each of the voters, a candidate, p , and a number $r \in [0, 1]$. We are asked if the probability of p winning in the original election (where S' is the set of votes cast) is greater than r .*

Definition 10 (Weighted Extension-bribery). *We are given a set S of votes which are possibly top-truncated, the weights of each of the voters in V , a collection $\Delta = (\delta_1, \dots, \delta_n)$ of extension-bribery cost functions, a preferred candidate, p , and a budget, B . We are asked if there exists an extension to the votes in S with cost $\leq B$ such that p is the winner.*

Informally, the Weighted Extension-bribery problem essentially tries to figure out if it is possible to extend a partially specified vote—as in, if it is possible to increase the number of ranked candidates specified in the preference order by suitably filling it up—so as to ensure that a preferred candidate, p , wins.

Now, we show that the Evaluation under Top-truncated Uncertainty problem with $r = 0$ (henceforth also referred to as the Evaluation problem) is equivalent to a special case of the Weighted Extension-bribery problem namely, Weighted Extension-bribery with zero costs (i.e. when $\delta_i = 0$, $1 \leq i \leq n$).

Theorem 16. *For a given voting protocol, Weighted Extension-bribery with zero costs is \mathcal{NP} -hard if and only if the Evaluation under Top-truncated Uncertainty problem (with $r = 0$) is \mathcal{NP} -hard.*

Proof. Consider an arbitrary instance of the Weighted Extension-bribery with S and p , and the same instance for the Evaluation problem. Now, it is clear that if there exists an extension in Weighted Extension-bribery problem, then p wins with probability greater than zero in the Evaluation problem. Conversely, if p wins with non-zero probability in the Evaluation problem, then this implies that there is at least one extension where it wins. \square

Next, before we look at the complexity of Weighted Extension-bribery with zero costs (and hence of Evaluation) for all the voting rules considered in this chapter, consider the following version (CWCM*) of CWCM with top-truncated votes where the only difference is that here we make it necessary for the non-manipulators to always have complete ballots (i.e. in CWCM* only the manipulators can cast top-truncated votes).

Definition 11 (CWCM*). *CWCM* with top-truncated votes is the same problem as CWCM with top-truncated votes, with the additional restriction that the non-manipulators always have complete preference orders.*

The first thing to observe here is that all the \mathcal{NP} -complete results for CWCM with top-truncated votes from Section 3.1 hold for CWCM* as well, since a close look at the proofs for Theorem 2, Theorem 3, Theorem 8, and Theorem 10 reveal that in all the cases we showed reductions to instances which always had complete orders for the non-manipulators. Hence, we have the following theorem.

Theorem 17. *CWCM* with top-truncated votes is \mathcal{NP} -complete for 3-candidate X_{\downarrow}^1 , 3-candidate X_{av}^2 , eliminate(X^3), 3-candidate Copeland $^\alpha$ for $\alpha \in [0, 1)$, and 3-candidate plurality with runoff rule, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

The second observation to make regarding CWCM* with top-truncated votes is that it is a special case of Weighted Extension-bribery with zero costs, where some the ballots are complete (those of the non-manipulators) and some are empty (those of the manipulators). Therefore, any hardness result for CWCM* carries over for Weighted Extension-bribery with zero costs (and also for the Evaluation problem as a consequence of Theorem 16), and so we have the following theorem.

Theorem 18. *Weighted extension-bribery problem with zero costs (and hence even the Evaluation problem) is \mathcal{NP} -complete for 3-candidate X_{\downarrow}^1 , 3-candidate X_{av}^2 , eliminate(X^3), 3-candidate plurality with runoff, and 3-candidate Copeland $^{\alpha}$ for $\alpha \in [0, 1)$, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Additionally, we also have the following result which shows that Weighted Extension-bribery with zero costs is in \mathcal{P} for certain voting rules considered here.

Theorem 19. *Weighted Extension-bribery with zero costs (and hence even the Evaluation problem) is in \mathcal{P} for X_{\uparrow} , plurality $_{\downarrow}$, veto $_{\downarrow}$, plurality $_{av}$, and the maximin protocol, where X is any positional scoring rule.*

Proof. For all the above mentioned protocols except plurality $_{\downarrow}$, the best we can do is to extend each top-truncated vote by placing p at its end (i.e. place p as k th candidate if $k - 1$ candidates are already ranked), if it isn't already present. In case of plurality $_{\downarrow}$, the best strategy is to complete each of the top-truncated votes by placing p at the topmost position possible followed by all the other as-yet unranked candidates in any arbitrary order. \square

Next we show that if Evaluation is hard then constructive manipulation with even a single manipulator is hard.

Definition 12 (CWIM-TTU). *In Constructive Weighted Individual Manipulation under Top-truncated Uncertainty (CWIM-TTU), we are given a set, S , of partially-revealed top-truncated ballots of the non-manipulators (which in turn may contain complete or top-truncated ballots themselves), the weight of the manipulator, a preferred candidate, p , and a number $r \in [0, 1]$. We are asked if the manipulator can cast his vote in such a way so as to ensure p wins with a probability greater than r .*

Theorem 20. *If Evaluation under Top-truncated Uncertainty is \mathcal{NP} -hard for a given protocol with k candidates, then CWIM-TTU with top-truncated votes is also \mathcal{NP} -hard for it with k candidates.*

Proof. Construct an instance of CWIM-TTU from an Evaluation instance by just adding a manipulator of weight 0. \square

Combining Theorem 18 and Theorem 20, we have the following result which says that for all the protocols considered in Section 3.1 for which CWCM with top-truncated ballots was hard, even individual manipulation with top-truncated votes is hard when there is uncertainty about the non-manipulators' votes.

Theorem 21. *CWIM-TTU with top-truncated votes is \mathcal{NP} -complete for 3-candidate X_{\downarrow}^1 , 3-candidate X_{av}^2 , $\text{eliminate}(X^3)$, 3-candidate plurality with runoff rule, and 3-candidate Copeland $^\alpha$ for $\alpha \in [0, 1)$, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Finally, we conclude this section by showing that the CWIM-TTU with top-truncated votes is in \mathcal{P} for $\text{eliminate}(\text{veto})$ (it is easy to see that it is also in \mathcal{P} for all the rules mentioned in Theorem 19). We show this by first proving that Weighted Extension-bribery with zero costs (and hence Evaluation) is in \mathcal{P} for $\text{eliminate}(\text{veto})$.

Theorem 22. *In the unique winner model, Weighted Extension-bribery with zero costs (and Evaluation) is in \mathcal{P} for $\text{eliminate}(\text{veto})$ when the number of candidates is bounded.*

Proof. Suppose there was an arbitrary extension W which resulted in p winning. Let the corresponding elimination order be $e = (c_1, c_2, \dots, c_m = p)$, where c_i is the candidate eliminated in the i th round. Now, we claim that the same elimination order can be achieved by doing the following:

- In each of the top-truncated votes, complete it by placing the candidates in the reverse order in which they appear in e . That is, place p if not already present, followed by c_{m-1} if not present, and so on until c_1 .

Doing the above results in the same elimination order as e . This can be shown through an inductive argument. When there are m candidates, c_1 which is eliminated first in e has been placed last wherever possible and this in turn will result in it getting eliminated first in our completion. Once c_1 is eliminated, we have $m - 1$ candidates with c_2 placed last wherever possible and therefore c_2 will be eliminated next. Continuing this way, guarantees the elimination order e .

To solve Weighted Extension-bribery, the campaign manager can try out all possible elimination orders, extend the votes as outlined above, and see if any of them results in

p winning. Doing so will only take polynomial time since the number of candidates are bounded. \square

Theorem 23. *In the unique winner model, CWIM-TTU with top-truncated votes (with $r = 0$) is in \mathcal{P} for $\text{eliminate}(\text{veto})$ when the number of candidates is bounded.*

Proof. Since the Evaluation problem is in \mathcal{P} , the manipulator can try out all possible complete orders to check if manipulation is possible. This is enough because, for $\text{eliminate}(\text{veto})$, any manipulation that can be achieved by top-truncated votes can be achieved by completing the vote appropriately (see Theorem 4). \square

The above result also implies that, in the unique winner model, CWCM under Top-truncated Uncertainty and with top-truncated votes (with $r = 0$) is in \mathcal{P} for $\text{eliminate}(\text{veto})$ when the number of candidates is bounded, since in $\text{eliminate}(\text{veto})$ any manipulation that can be induced by an arbitrary set of the manipulators' vote can be induced if all the manipulators vote in the same way [11, Lemma 12].

3.3.2 When there is only probabilistic information on the non-manipulators

Manipulation under probabilistic uncertainty was introduced by Conitzer *et al.* [13]. In their work they introduced the Weighted Evaluation problem which, given a probability distribution on the votes, and a number $r \in [0, 1]$, asks if the probability of a candidate winning is greater than r . Subsequently, they also proved that if CWCM (with complete votes) for a voting protocol is hard then so is Weighted Evaluation [13, Theorem 15]. Now, since we additionally allow top-truncated votes, we can state the following result which is almost equivalent to [13, Theorem 15] with the only difference being that the “set of all possible votes” would now contain top-truncated votes as well and that the reduction here is from CWCM with top-truncated votes. In fact, most of the results in this section are only extensions to the corresponding result from Conitzer *et al.*'s paper [13] that arise as a result of allowing top-truncated voting.

Theorem 24. *If CWCM with top-truncated votes is \mathcal{NP} -hard for a given protocol with k candidates, then Weighted Evaluation when top-truncated votes are allowed is also \mathcal{NP} -hard for it (with k candidates) even if $r = 0$, the votes are drawn independently, and only the following types of distributions are allowed: (1) the vote's distribution is uniform over all possible votes, or (2) the vote's distribution puts all of the probability mass on a single vote.*

Proof. We can proceed exactly as in [13, Theorem 15] to prove this result. To begin, consider an arbitrary instance of CWCM with top-truncated votes with the set S of non-manipulators, T of manipulators, and p , the preferred candidate. Let the Weighted Evaluation instance be the same, with $r = 0$, and p being the candidate to be evaluated. Now, for the voters in S , since their votes are already given in the CWCM instance, in the Weighted Evaluation instance its distribution places all of the probability mass on that given vote. On the other hand, if the voter was in T , then its vote is drawn uniformly from the set of all the possible votes. Now, it is easy to see that p wins the probability greater than zero in the Weighted Evaluation instance if and only if there exists some way to cast votes for voters in T such that there will a successful constructive manipulation in favour of p in the CWCM instance. \square

Corollary 25. *Weighted Evaluation when top-truncated votes are allowed is \mathcal{NP} -hard for 3-candidate X^1_{\downarrow} , 3-candidate X^2_{av} , eliminate(X^3), 3-candidate plurality with runoff, and 3-candidate Copeland $^\alpha$ for $\alpha \in [0, 1)$, when the votes are drawn independently, and the distributions allowed are: (1) uniform over all possible votes, or (2) the vote's distribution puts all of the probability mass on a single vote, where X^1 represents all scoring rules except plurality and veto, X^2 represents all scoring rules except plurality, and X^3 represents all scoring rules except veto.*

Next, we show a relation between the Weighted Evaluation and manipulation with a single manipulator as in [13]. Note that the difference between CWIM with Uncertainty (CWIM-U) that we define below and CWIM-TTU that we defined in the Section 3.3.1 is in how the partial information about the non-manipulators' votes are specified.

Definition 13 (CWIM-U). *In Constructive Weighted Individual Manipulation under Uncertainty (CWIM-U), given a distribution over all the non manipulators' votes, the weights of the non-manipulators, the weight of the manipulator, a preferred candidate, p , and a number $r \in [0, 1]$, we are asked if the manipulator can cast his vote in such a way so as to ensure p wins with a probability greater than r .*

Theorem 26. *If Weighted Evaluation with top-truncated votes is \mathcal{NP} -hard for a protocol with k candidates and some restrictions on the distribution, then CWIM-U with top-truncated votes is also \mathcal{NP} -hard for it with k candidates and the same restrictions on the distribution.*

Proof. Construct an instance of CWIM-U from an arbitrary Weighted Evaluation instance by just adding a manipulator of weight 0. \square

Finally, we show that Weighted Evaluation when top-truncated votes are allowed can be hard even if CWCM with top-truncated votes is in \mathcal{P} . To do so, we consider eliminate(veto) for which CWCM with top-truncated votes was shown to be in \mathcal{P} for bounded number of candidates (see Corollary 5).

Theorem 27. *In 3-candidate eliminate(veto) , Weighted Evaluation when top-truncated votes are allowed is \mathcal{NP} -hard even if $r = 0$, the votes are drawn independently, and the distribution over each vote has a positive probability for at most 2 of the votes.*

Proof. We show this by a reduction from an arbitrary instance of the Partition problem to the following instance of Weighted Evaluation. Let a, b , and p be three candidates. Let there be a vote of weight 1 for $(p > a > b)$. For each k_i in the partition instance, let it have a weight k_i and vote for $(a > p > b)$ and $(b > p > a)$ with probability $1/2$ each.

Now, we can see that p wins if and only if $(a > p > b)$ and $(b > p > a)$ are voted by exactly K of the vote weight, because failing to do so would mean that p will be eliminated in the second round. But then this is possible if and only if there exists a partition. \square

3.4 Conclusion

In this chapter, we studied the problem of manipulation of weighted elections when the agents are allowed to specify top-truncated preferences and also looked at the impact on complexity of manipulation when there is uncertainty about the non-manipulators' votes. Regarding the first problem, we provided general results for constructive and destructive manipulation in all scoring rules, elimination versions of all scoring rules, the plurality with runoff rule, a family of election systems known as Copeland $^\alpha$, and the maximin protocol. These results are summarized in Table 3.1. As was also noted by Narodytska and Walsh in their study of manipulation with top-truncated votes for Borda, STV, and Copeland $^{0.5}$ [38], there are three broad trends that we can observe. First is the case where top-truncated voting has a strong impact on manipulation and it in turn results in a decrease in the worst-case complexity of manipulation as compared to the complete votes case. Examples of this are all the scoring rules when using the round-up evaluation scheme and the maximin rule. Second is the case where top-truncated voting has some impact on manipulation and in fact even causes more strategic voting, but yet the worst-case complexity of manipulation remains the same as compared to the complete votes case. Some examples of voting rules which fall into this category are the Copeland $^\alpha$, X_\downarrow for any scoring rule X that is not isomorphic to plurality or veto etc. Lastly, we also see that there are voting rules for

Voting Rule	CWCM (#cand)	DWCM (#cand)
X_{\uparrow}	\mathcal{P}	\mathcal{P}
Plurality $_{\downarrow}$	\mathcal{P}	
Veto $_{\downarrow}$	\mathcal{P}	
X^1_{\downarrow}	<i>\mathcal{NP}-complete (3)</i>	
Plurality $_{av}$	\mathcal{P}	
X^2_{av}	<i>\mathcal{NP}-complete (3)</i>	\mathcal{P}
eliminate(Veto)	\mathcal{P}	
eliminate(X^3)	<i>\mathcal{NP}-complete (3)</i>	
Plurality with runoff	<i>\mathcal{NP}-complete (3)</i>	
<i>Copeland$^{\alpha}$, $\alpha \in [0, 1)$</i>	<i>\mathcal{NP}-complete (3)</i>	
Maximin	\mathcal{P}	\mathcal{P}

X : All scoring rules

X^1 : All scoring rules except plurality and veto

X^2 : All scoring rules except plurality

X^3 : All scoring rules except veto

Table 3.1: Complexity of CWCM and DWCM with Top-truncated Votes. The entries in bold indicate that there is a change in complexity of CWCM as compared to the case of complete votes, the non-highlighted ones indicate that there is no change to the worst-case complexity as compared to the case of complete votes, and the italicized entries indicate those rules for which there is more opportunity for manipulation but for which the worst-case complexity is still the same as compared to the case of complete votes.

which top-truncated voting has no impact whatsoever on strategic voting. For instance, top-truncated voting has no impact on STV and eliminate(veto).

Following this, we ended this chapter by exploring the second problem i.e. when there is uncertainty about the non-manipulators' votes. Here we discussed two possible ways in which the uncertainty can be modeled and we also showed that in both cases even individual manipulation under uncertainty was hard when constructive coalitional manipulation was hard. To the best of our knowledge, we are the first to study the impact on manipulation with top-truncated ballots when there is uncertainty about the non-manipulators' votes.

Chapter 4

Reinstating Combinatorial Protections for Manipulation and Bribery in Single-Peaked and Nearly Single-Peaked Electorates

In the previous chapter we looked at the complexity of manipulation when the voters are allowed to cast top-truncated ballots. The fundamental assumption in all of the results was that there is no particular structure in the agents' preferences, and so, given the set of candidates, C , it was assumed that an agent can specify any linear order over C or a non-empty subset of it. However, there are many election scenarios where the preferences of agents have an underlying structure, meaning that there is a subset of admissible votes that are never cast. For instance, there have been studies that have shown that many important election scenarios like the U.S. presidential elections and elections in committees are close to having a single-peaked structure (see, for example, [8], [39]).

One natural question that arises in the above context is: “What happens to the host of results in computational social choice where \mathcal{NP} -hardness shields were obtained for different voting rules using constructions on combinatorially rich structures such as partitions and covers? Do they still hold when the electorate has structured preferences?”. This question was first raised by Walsh [43] and was subsequently studied in detail by Faliszewski *et al.* for manipulation and control in single-peaked electorates [27], by Brandt *et al.* for bribery in single-peaked electorates [9], and by Faliszewski *et al.* [24] for manipulation, bribery, and control when the electorate is nearly single-peaked. The overarching

theme in all these papers was that the previously obtained combinatorial protections vanish when the voters have structured preferences—i.e. many of the \mathcal{NP} -hardness results fall into polynomial time.

In light of the above results, our results in this chapter take this line of research in a new direction by looking at the impact of partial preferences on manipulative actions in single-peaked and nearly single-peaked electorates. In particular, we consider top-truncated preferences, and we look at their impact on manipulative actions in single-peaked and nearly single-peaked settings. In doing so, we arrive at a number of surprising results, which in turn forms the theme of this chapter—of reinstating combinatorial protections by allowing top-truncated voting. Although, as noted by Faliszewski *et al.* [24], polynomial time algorithms for manipulative actions such as bribery and control are not always unethical or “bad” (as it can be a valuable tool in the hands of, say, the campaign manager, committee chair etc.), in the majority of situations they aren’t socially “good” either and hence we believe that it is important to have elections protected against them. This thought, along with the “easiness” results previously obtained, and the fact that, among structured preference profiles, single-peaked and nearly single-peaked preferences are the most widely studied, forms the main motivation of our work here. The overarching theme in this chapter is that top-truncated voting is useful in reinstating combinatorial protections in single-peaked and nearly single-peaked electorates, and we believe that these results form a *win-win* scenario: allowing voters to specify top-truncated ballots (or partial preferences, in general) is extremely useful and often necessary in many multi-agent systems applications and even in real-world elections, and allowing this additional flexibility in turn gives us what we want in terms of making the complexity of many manipulative-action problems hard.

The rest of this chapter is organized as follows. In Section 4.1 we outline our contributions. In Section 4.2 we present all our results for manipulation when the electorate is both single-peaked and nearly single-peaked. Following this we turn to the problem of weighted bribery in Section 4.3. Finally, in Section 4.4, we address the question on whether our observations—of top-truncated voting reinstating combinatorial protections—are universal and ask if there are voting rules for which top-truncated voting is not beneficial in terms of increasing the complexity of manipulation.

4.1 Contributions

We, for the first time, systematically study the impact of partial voting on manipulative actions in structured preference profiles. In particular, we look at the problem of manipu-

lation and bribery in single-peaked and nearly single-peaked settings when top-truncated ballots are allowed.

Second, under the assumption that the voters submit complete ballots, we first provide polynomial time algorithms for manipulation and weighted-bribery for certain voting rules in single-peaked and nearly single-peaked settings, thus extending the works of Faliszewski *et al.* for manipulation [27], Brandt *et al.* for bribery [9], and Faliszewski *et al.* for nearly single-peaked electorates [24]. We then show how these polynomial-time problems become \mathcal{NP} -complete when top-truncated ballots are allowed.

Third, we show an example of a natural voting rule where, contrary to intuition, the complexity of manipulation actually increases when moving from the general case (i.e. when there is no restriction on the preferences) to the single-peaked case. In particular, in Theorem 32 we show how the complexity of manipulating eliminate(veto), when top-truncated ballots are allowed, moves from being in \mathcal{P} in the general case to being \mathcal{NP} -complete in the single-peaked case.

4.2 Manipulation

In this section we study CWCW with top-truncated votes in both single-peaked and nearly single-peaked electorates. Since the theme here is the reinstatement of combinatorial protections by top-truncated voting, for all the voting rules considered in this section, we present both the “easiness” result (if not already known from previous work) as well as the subsequent “hardness” result that arises as a consequence of allowing top-truncated ballots.

4.2.1 Single-Peaked Electorates

Walsh was the first to consider manipulation with single-peaked preferences and he showed that STV remains \mathcal{NP} -hard to manipulate for 3 candidates [43]. Subsequently, Faliszewski *et al.* showed that for many voting protocols which are usually hard to manipulate, restricting the preferences to being single-peaked makes them easy [27]. In particular, they showed that any 3-candidate scoring rule with $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ is easy to manipulate. This result was then extended to obtain a complete characterization for any m -candidate scoring rule by Brandt *et al.* [9]. Here, we look at 3-candidate scoring rules again and study the impact on complexity of manipulation when top-truncated voting is allowed. The following

results for the case of 3-candidate Borda rule was also independently shown by Fitzsimmons and Hemaspaandra [29]. Additionally, we also note that the proofs for Theorem 28 and Theorem 29 in this chapter are very similar to the ones for Theorem 2 and Theorem 3 in Chapter 3.

Theorem 28. *For any 3-candidate scoring rule X that is not isomorphic to plurality or veto, in single-peaked electorates, CWCM with top-truncated votes in X_\downarrow is \mathcal{NP} -complete.*

Proof. Since there are only three candidates, the scoring vector for the corresponding positional scoring rule is defined by $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$, where $\alpha_1 > \alpha_2 > \alpha_3 = 0$ (because $\alpha_1 = \alpha_2$ is isomorphic to veto, $\alpha_2 = \alpha_3$ is isomorphic to plurality, and α_3 can be taken to be zero since translating the scores in a scoring rule does not affect the outcome of the rule).

It is easy to see that the problem is in \mathcal{NP} . To prove \mathcal{NP} -hardness, we proceed by considering the following three cases. In all the cases we use a reduction from the Partition problem.

Case 1: $\alpha_2 < \alpha_1 < 2\alpha_2$. Consider the following instance of CWCM, where a, b , and p are the three candidates. Let $p \prec a \prec b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(p > a > b), (a > p > b), (a > b > p), (b > a > p), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S , let there be a vote of weight K voting $(a > b > p)$, $(b > a > p)$, $(b > a \sim p)$, and $(p > a \sim b)$ each. As a result the scores of a, b , and p are $\alpha_1 K + \alpha_2 K$, $\alpha_1 K + 2\alpha_2 K$, and $\alpha_2 K$ respectively. In T , let each k_i have a vote of weight k_i .

Suppose there exists a partition. Let those manipulators in the first half vote $(p > a > b)$ and let the others vote $(p > a \sim b)$. Then the scores of a, b , and p are $\alpha_1 K + 2\alpha_2 K$, $\alpha_1 K + 2\alpha_2 K$, and $\alpha_1 K + 2\alpha_2 K$ respectively, and hence p wins by tie-breaking.

Conversely, suppose there exists a manipulation in favour of p . Let x and y denote the total weight of the manipulators voting $(p > a > b)$ and $(p > a \sim b)$, respectively. Since there exists a successful manipulation in favour of p , the score of p should be at least as much as that of a . Therefore, we have: $x\alpha_1 + y\alpha_2 + \alpha_2 K \geq \alpha_1 K + x\alpha_2 + \alpha_2 K$. Using the fact that $x + y = 2K$, this simplifies to $(x - K)(\alpha_1 - 2\alpha_2) \geq 0$. But since we assumed that $\alpha_1 < 2\alpha_2$, we have $x \leq K$. Similarly, the score of p should also be at least as much as b and so we have $x\alpha_1 + y\alpha_2 + \alpha_2 K \geq \alpha_1 K + 2\alpha_2 K$. Simplifying this we have $(x - K)(\alpha_1 - \alpha_2) \geq 0$, and since we assumed that $\alpha_2 < \alpha_1$, therefore $x \geq K$. But then, x cannot be both greater than and lesser than K at the same time. So x has to equal to K and this in turn implies that $y = K$ and that there exists a partition.

Case 2: $\alpha_1 > 2\alpha_2$. Consider the following instance of CWCM, where a, b , and p are the three candidates. Let $a L p L b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S , let there be a vote of weight $(2\alpha_1 - \alpha_2)K$ voting $(a > p > b)$, and $(b > p > a)$ each. As a result the scores of a, b , and p are $(2\alpha_1 - \alpha_2)\alpha_1 K$, $(2\alpha_1 - \alpha_2)\alpha_1 K$, and $2(2\alpha_1 - \alpha_2)\alpha_2 K$ respectively. In T , let each k_i have a vote of weight $(\alpha_1 - 2\alpha_2)k_i$.

Suppose there exists a partition. Let those manipulators in the first half vote $(p > a > b)$ and let the others vote $(p > b > a)$. Then the scores of a, b , and p are all $2(\alpha_1^2 - \alpha_2^2)K$, and hence p wins by tie-breaking.

Conversely, suppose there exists a manipulation in favour of p . Let x, y , and z be the sum of the k_i 's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively. Since there exists a successful manipulation in favour of p , the score of p should be at least as much as that of a . Therefore, we have: $2(2\alpha_1 - \alpha_2)\alpha_2 K + (x + y)\alpha_1(\alpha_1 - 2\alpha_2) + z\alpha_2(\alpha_1 - 2\alpha_2) \geq (2\alpha_1 - \alpha_2)\alpha_1 K + \alpha_1 K + x\alpha_2(\alpha_1 - 2\alpha_2)$. Using the fact that $x + y + z = 2K$, this simplifies to

$$\alpha_2 K - z(\alpha_1 - \alpha_2) \geq x\alpha_2. \quad (4.1)$$

Doing the same with respect to p and b we have,

$$\alpha_2 K - z(\alpha_1 - \alpha_2) \geq y\alpha_2. \quad (4.2)$$

Adding (4.1) and (4.2) we have $z(2\alpha_1 - 3\alpha_2) \leq 0$. Since we assumed that $\alpha_1 > 2\alpha_2$, $(2\alpha_1 - 3\alpha_2) > 0$ and this in turn implies that $z \leq 0$. But then z cannot be less than zero and so it should be zero. Substituting $z = 0$ in (4.1) and (4.2) we have $x \leq K$, and $y \leq K$ respectively. Since $x + y + z = 2K$, this implies that $x = K$ and $y = K$ and that there exists a partition.

Case 3: $\alpha_1 = 2\alpha_2$. Consider the following instance of CWCM, where a, b , and p are the three candidates. Let $a L p L b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S , let there be a vote of weight $3K$ voting $(a > b \sim p)$, and $(b > a \sim p)$ each. As a result the scores of a, b , and p are $3\alpha_2 K$, $3\alpha_2 K$, and 0 respectively. In T , let each k_i have a vote of weight k_i .

Suppose there exists a partition. Let those manipulators in the first half vote $(p > a > b)$ and let the others vote (p, b, a) . Then the scores of a, b , and p are $4\alpha_2 K$, $4\alpha_2 K$, and

$2\alpha_1 K$, respectively. Since $\alpha_1 = 2\alpha_2$, all of them have the same score and hence p wins by tie-breaking.

Conversely, suppose there exists a manipulation in favour of p . Let x, y , and z be total weight of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively. Since there exists a successful manipulation in favour of p , the score of p should be at least as much as that of a . Therefore, we have: $(x + y)\alpha_1 + z\alpha_2 \geq 3\alpha_2 K + x\alpha_2$. Using the fact that $x + y + z = 2K$, and that $\alpha_1 = 2\alpha_2$ this simplifies to $y \geq K$. Doing the same with respect to p and b we have, $x \geq K$. But then since $x + y + z = 2K$, this implies that $x = K$ and $y = K$ and that there exists a partition. \square

Theorem 29. *For any 3-candidate scoring rule X that is not isomorphic to plurality, in single-peaked electorates, CWCM with top-truncated votes in X_{av} is \mathcal{NP} -complete.*

Proof. Clearly, this problem is in \mathcal{NP} . To show \mathcal{NP} -hardness, we reduce from an arbitrary instance of the Fixed-difference Subset Sum problem.

Consider the following instance of CWCM, where a, b , and p are the three candidates. Let $a \succ p \succ b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S , let there be a vote of weight $8K$ voting $(a > b \sim p)$, and $(b > a \sim p)$ each, vote of weight $3K$ voting $(p > b > a)$, and a vote of weight K voting $(p > a > b)$. As a result the scores of a, b , and p are $8\alpha_1 K + 5\alpha_2 K$, $8\alpha_1 K + 7\alpha_2 K$, and $4\alpha_1 K + 8\alpha_2 K$ respectively. In T , let each k_i have a vote of weight $2k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. Let those manipulators who are in S_1 vote (p, a, b) , those in S_2 vote $(p > b > a)$, and let all the remaining manipulators vote $(p > a \sim b)$. If x, y , and z denote the sum of k_i 's of the manipulators who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of a, b , and p are $8\alpha_1 K + 5\alpha_2 K + 2x\alpha_2 + z\alpha_2$, $8\alpha_1 K + 7\alpha_2 K + z\alpha_2$, and $8\alpha_1 K + 8\alpha_2 K$, respectively. Now, since $x - y = K$ and $x + y + z = 2K$, we see that scores of a, b , and p are the same and hence p wins by tie-breaking.

Conversely, suppose there existed a successful manipulation for p . From above we know that if x, y , and z denote the sum of k_i 's of the manipulators who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then the scores of a, b , and p are $8\alpha_1 K + 5\alpha_2 K + 2x\alpha_2 + z\alpha_2$, $8\alpha_1 K + 7\alpha_2 K + z\alpha_2$, and $8\alpha_1 K + 8\alpha_2 K$, respectively. Since there is a successful manipulation the score of p should be at least as much as that of a , and so this in turn implies that $x - y \leq K$. Similarly, doing it with respect to p and b we have $x - y \geq K$. But then $x - y$ cannot be both greater and smaller than K at the same time. So this implies that

$x - y = K$ and that there exists two subsets such that $\sum S_1 - \sum S_2 = K$, where $x = \sum S_1$ and $y = \sum S_2$. \square

From Theorem 28 and Theorem 29, we can see that a relaxation of the complete votes assumption by additionally allowing top-truncated votes actually increases the complexity of CWCM for all 3-candidate scoring rules with $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ from being in \mathcal{P} [27] to being \mathcal{NP} -complete when either the round-down or average score evaluation schemes are used. However, with the round-up evaluation scheme manipulation become easy for all m -candidate scoring rules as shown below.

Theorem 30. *In single-peaked electorates, computing if a coalition of manipulators can manipulate $\text{plurality}_\downarrow$, veto_\downarrow , plurality_{av} , and X_\uparrow , for any scoring rule X , with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Proof. For all the voting rules except $\text{plurality}_\downarrow$, the manipulators can simply check if all of them voting ($p > a \sim b$) will make p a winner. If not they cannot make p a winner. In case of $\text{plurality}_\downarrow$, all the manipulators can vote for any single-peaked order consistent with the societal order L that has p at the top. \square

Another interesting point to note here is that Theorem 28, Theorem 29, and Theorem 30 together also imply that the restriction of preferences to being single-peaked has no effect on the complexity of manipulation with top-truncated ballots, since, as we have already seen, the same corresponding results were obtained in Chapter 3.

Next, we look at CWCM in Copeland $^\alpha$ and we present both the “easiness” and the “hardness” result.

Theorem 31. *In single-peaked electorates, for 3-candidate Copeland $^\alpha$, $\alpha \in \mathbb{Q}$, $0 \leq \alpha < 1$,*

1. *CWCM with **complete votes** is in \mathcal{P} .*
2. *CWCM with **top-truncated votes** is \mathcal{NP} -complete.*

Proof. We first prove the polynomial time result for CWCM with complete votes and then show the \mathcal{NP} -completeness of CWCM with top-truncated votes.

Let a, b , and p , be the three candidates. Without loss of generality, we need to consider only the following two linear orderings L :

- i) $p \ L \ a \ L \ b$: In this case, the only strategy for all the manipulators is to just vote ($p > a > b$).

- ii) $a L p L b$: In this case, the linear ordering restricts the set of allowed votes to $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a)\}$. As a result, it can be seen that regardless of the weights of the voters in S , p cannot lose to both a and b in a pairwise election. Therefore, the only possibilities are p losing to a (respectively, b), but winning against b (a), or p tying against both a and b . In the latter scenario all the manipulators can either vote $(p > a > b)$ or $(p > b > a)$, while in the former all of them can vote $(p > b > a)$ (respectively, $(p > a > b)$).

For the second part, it is easy to see that the problem is in \mathcal{NP} . To prove \mathcal{NP} -hardness, we reduce from an arbitrary instance of the Fixed-difference Subset Sum problem.

Consider the following instance of CWCW, where a, b , and p are the three candidates. Let $a L p L b$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a > p > b), (p, a, b), (p > b > a), (b > p > a), (a > b \sim p), (b > a \sim p), (p > a \sim b)\}$. In S , let there be a vote of weight $6K$ voting $(a > b \sim p)$ and $(b > a \sim p)$ each, and a vote of weight $2K$ voting $(p > a > b)$. In T , let each k_i have a vote of weight $2k_i$.

Suppose there exists S_1, S_2 such that $\sum S_1 - \sum S_2 = K$. In Copeland^a, it can be assumed that all the manipulators rank p first. So, let the manipulators in S_1 vote $(p > b > a)$, those in S_2 vote $(p > a > b)$, and let the rest vote $(p > a \sim b)$. If $N_V(r, s)$ denotes the total number of votes in V which rank r prior to s , and $D_V(r, s) = N_V(r, s) - N_V(s, r)$, then $D_{S \cup T}(p, a) = 0$ and $D_{S \cup T}(p, b) = 0$. Therefore, the score of p , $s(p) = 2\alpha$. Also since $\sum S_1 - \sum S_2 = K$, $D_T(a, b) = -2K$, while $D_S(a, b) = 2K$. Therefore, $D_{S \cup T}(a, b) = 0$ and so, both receive a score 2α . As a result, p wins by tie-breaking.

Conversely, suppose there exists a successful manipulation in favour of p . If x, y , and z , denote the sum of k_i 's of the manipulators in T who vote $(p > a > b)$, $(p > b > a)$, and $(p > a \sim b)$, respectively, then without taking into account the pairwise election between a and b in T , the score of p , a , and b is $2\alpha, 1 + \alpha$, and α , respectively. Now since $2\alpha < 1 + \alpha$ for all rational $\alpha \in [0, 1)$, therefore, the only way p would win this is if including the pairwise election between a and b in T results in a tie between them. So this implies that $D_{S \cup T}(a, b) = 2K + 2x - 2y = 0$, or $y - x = K$. \square

Our final result for manipulation under single-peaked preferences is the very interesting case of eliminate(veto).

Theorem 32. *In single-peaked electorates, and in the unique winner model, for eliminate(veto),*

1. CWCM with **complete votes** is in \mathcal{P} when the number of candidates is bounded.
2. CWCM with **top-truncated votes** is \mathcal{NP} -complete for even three candidates.

Proof. We first present the polynomial time algorithm for CWCM with complete votes.

Suppose there was an arbitrary set W of the manipulators' votes which—along with S —resulted in p winning. Let the corresponding elimination order be $e = (c_1, \dots, c_m = p)$. To prove this theorem, we need the following lemmas.

Lemma 3. *During each round of the eliminate(veto), only the rightmost or the leftmost candidate in the linear ordering L is eliminated.*

Proof. Let $c_1 L c_2 L \dots L c_m$ be any arbitrary linear ordering. It is clear that in the first round of eliminate(veto) only c_1 or c_m will be eliminated since all the votes only have either of them placed at the end. Also, since the votes are single-peaked, for any candidate c_j , the only candidates which can be immediately on top of it in a vote where c_j is placed last are its right and left neighbours, where the right neighbour of the rightmost candidate is the leftmost one and vice-versa. Therefore, in particular, if c_1 is the candidate eliminated in the first round then all its last votes gets transferred to either c_2 or c_m . Hence in the subsequent round only one of c_2 or c_m will be eliminated. Continuing this way we can see that at every round r only one of the corner candidates in L (which now is the linear ordering over the remaining $m - r + 1$ candidates) will be eliminated. \square

Lemma 4. *The reverse of e is a single-peaked order with respect to the given linear order L .*

Proof. Our claim is that $O : (p = c_m) > c_{m-1} > \dots > c_1$ is a valid single-peaked order with respect to L . To prove this, let us assume the contrary. This implies that there exists three candidates c_i, c_j, c_k such that $c_j >_O c_i >_O c_k$, but according to the linear ordering L it is only possible that $c_i L \dots L c_k L \dots L c_j$ or $c_j L \dots L c_k L \dots L c_i$. Now, $c_j >_O c_i >_O c_k$ implies that, among c_i, c_j , and c_k , c_k is eliminated first, followed by c_i , and then c_j . But then, since at any round r only the corner elements can be eliminated (Lemma 3), c_k can never be eliminated before both c_i and c_j in either of the two linear orderings. Hence e cannot be a valid elimination order if its reverse is not single-peaked. \square

Proof of Theorem 32 contd. Now since the reverse of every elimination order e caused by any arbitrary manipulation in favour of p is single-peaked (Lemma 4), the manipulators can try out all possible single-peaked orders which have p at the top and can try to induce e by collectively voting for the same. This is enough because we know that in

eliminate(veto) any elimination order that is achieved by an arbitrary set of manipulators' votes can be achieved if all the manipulators vote the same way (by using the reverse of the elimination order) [11, Lemma 12]. And since, when the number of candidates are bounded, there are only polynomial number of single-peaked orders which have p at the top, this can be done in polynomial time.

We now go on to show the hardness of CWCM with top-truncated votes for 3-candidate eliminate(veto). Clearly, the problem is in \mathcal{NP} . To show \mathcal{NP} -hardness, we reduce from an arbitrary instance of the Partition problem.

Consider the following instance of CWCM, where a, b , and p are the three candidates. Let $a \succ b \succ p$ be the linear ordering over the candidates. This linear ordering in turn restricts the set of allowed votes to $\{(a \succ b \succ p), (b \succ a \succ p), (b \succ p \succ a), (p \succ b \succ a), (a \succ b \sim p), (b \succ a \sim p), (p \succ a \sim b)\}$. In S , let there be a voter of weight $K + 2$ voting $(b \succ p \succ a)$, voter of weight $K - 1$ voting $(b \succ a \succ p)$, a voter of weight 1 voting $(a \succ b \succ p)$, a voter of weight 2 voting $(p \succ a \sim b)$, and a voter of weight 3 voting $(a \succ b \sim p)$. In T , let each k_i have a vote of weight k_i .

The first thing to observe is, p cannot be unique winner if a is eliminated in the first round. This is because once a is eliminated, $score(b) = score(p) = 2K + 2$. Therefore, for p to be a unique winner, b has to be eliminated in the first round. This in turn implies that the total score of a and p should be at least 1 more than that of b (since we are in the unique winner model). So we have,

$$\begin{aligned} score_T(a) + K + 3 &\geq 2K + 3 + score_T(b) \implies score_T(a) - score_T(b) \geq K \\ score_T(p) + K + 4 &\geq 2K + 3 + score_T(b) \implies score_T(p) - score_T(b) \geq K - 1. \end{aligned}$$

where $score_T(p)$ is the score p receives from the votes in T .

Now, it is possible to show that the above equations can be satisfied only if b receives a score of 0 from T . Therefore, this implies that all the votes are concentrated on $(p \succ a \sim b)$ and $(a \succ b \sim p)$. As a result, given the equations above, we basically have two possibilities: a total weight of $K + 1$ voting for $(a \succ b \sim p)$, $K - 1$ voting $(p \succ a \sim b)$, or a total weight of K voting $(a \succ b \sim p)$ and $(p \succ a \sim b)$ each. In the first case it is easy to see that p will be eliminated in the second round, while in the latter case p will be a unique winner. Hence p can be a unique winner if and only if there exists a partition. \square

Theorem 32 is most interesting not because of the fact that it follows our theme of top-truncated voting reinstating combinatorial protections, but for the following other reasons. First is the very unusual behaviour that it is showing. Eliminate(veto), when there are

only a bounded number of candidates and in the unique winner model, is known to be in \mathcal{P} for practically everything—from CWCM with complete votes in the general case [11], to CWCM with top-truncated votes in the general case (see Corollary 5 in Chapter 3), and even when there is only partial information (in the form of top-truncated votes) on the non-manipulators’ votes (see Theorem 23 in Chapter 3 and the discussion following that). However, here, with single-peaked preferences and with top-truncated votes, it is \mathcal{NP} -complete even when there are only three candidates. Second, what makes Theorem 32 even more interesting is the fact that this actually serves as a counterexample (with the only caveat that, to be fair, they had considered only complete votes in that paper) to a conjecture stated by Faliszewski *et al.* [27] where they say that they do not expect the complexity of manipulation for “any existing, natural voting system” to increase when moving from the general case (where there is no restriction on the preferences) to the single-peaked case. But this is exactly what we are seeing here.

4.2.2 Nearly Single-Peaked Preferences

For nearly single-peaked electorates, Faliszewski *et al.* were the first to look at the complexity of manipulation, bribery, and control [24]. In that paper, they introduced several notions of nearness and among them was the k -maverick-SP-society where all but at most k of the voters are consistent with the societal order L . As noted before, here we only consider this notion of “nearness”. We start off by looking at 3-candidate scoring rules and we show the impact of top-truncated voting on CWCM. Note that Faliszewski *et al.* showed that for all 3-candidate scoring rules that are not isomorphic to plurality CWCM for 1-maverick-SP-societies was \mathcal{NP} -complete [24]. The \mathcal{NP} -completeness proofs in this section are all based on the corresponding results for the case of single-peaked electorates in Section 4.2.1.

Theorem 33. *In 1-maverick-SP societies, for any 3-candidate scoring rule X that is not isomorphic to plurality or veto, CWCM with top-truncated votes in X_\downarrow is \mathcal{NP} -complete.*

Proof. Proceed the same way as in Theorem 28 by considering three cases.

Case 1: $\alpha_2 < \alpha_1 < 2\alpha_2$. Construct the following instance of CWCM with p L a L b as the linear ordering and with S having a vote of weight K voting $(a > b > p)$, $(b > p > a)$, $(a > b \sim p)$, and $(b > a \sim p)$ each. Note that here $(b > p > a)$ is the maverick. In T , let each k_i have a vote of weight k_i . Use Partition for reduction.

Case 2: $\alpha_1 > 2\alpha_2$. Construct the following instance of CWCM with $a L p L b$ as the linear ordering and with S having a vote of weight $(2\alpha_1 - \alpha_2)K$ voting for $(a > p > b)$, $(b > a > p)$ and $(b > a \sim p)$ each, and a vote of weight $2(2\alpha_1 - \alpha_2)K$ voting $(p > a \sim b)$. Note that here $(b > a > p)$ is the maverick. In T , let each k_i have a vote of weight $(\alpha_1 - 2\alpha_2)k_i$. Use Partition for reduction.

Case 3: $\alpha_1 = 2\alpha_2$. Construct the following instance of CWCM with $a L p L b$ as the linear ordering and with S having a vote of weight K voting $(b > a > p)$ and $(b > a \sim p)$ each, and a vote of weight $2K$ voting $(a > b \sim p)$. Note that here $(b > a > p)$ is the maverick. In T , let each k_i have a vote of weight k_i . Use Partition for reduction. \square

Theorem 34. *In 1-maverick-SP societies, for any 3-candidate scoring rule X that is not isomorphic to plurality, CWCM with top-truncated votes in X_{av} is \mathcal{NP} -complete.*

Proof. Proceed the same way as in Theorem 29. Construct the following instance of CWCM with $a L p L b$ as the linear ordering and with S having a vote of weight $2K$ voting $(a > b > p)$, $(b > p > a)$, $(a > b \sim p)$, and $(b > a \sim p)$ each. Note that here $(a > b > p)$ is the maverick. In T , let each k_i have a vote of weight $2k_i$. Use Fixed-difference Subset Sum for reduction. \square

Finally, it is also easy to see that the following theorem holds based on the algorithm given in Theorem 30.

Theorem 35. *In k -maverick-SP societies, computing if a coalition of manipulators can manipulate X_{\uparrow} , for any 3-candidate scoring rule X , plurality $_{\downarrow}$, veto $_{\downarrow}$, plurality $_{av}$ with weighted top-truncated votes takes polynomial time (for any number of candidates).*

Next, we look at eliminate(veto) and we show how top-truncated voting increases the complexity of manipulation for eliminate(veto) in 1-maverick-SP electorates and that it continues to portray the unusual behaviour noted earlier.

Theorem 36. *In 1-maverick-SP electorates, and in the unique winner model, for eliminate(veto),*

1. *CWCM with **complete votes** is in \mathcal{P} when the number of candidates is bounded.*
2. *CWCM with **top-truncated votes** is \mathcal{NP} -complete for even three candidates.*

Proof. We first show how CWCM with complete votes is in \mathcal{P} . To do the same, we will use the following lemma.

Lemma 5. *Any successful manipulation for p caused by an arbitrary set of votes of the manipulators can be achieved if all the manipulators vote for one of the single-peaked orders (that is consistent with the input linear ordering L) that has p at the top.*

Proof. Suppose there was an arbitrary set of votes W which resulted in a successful manipulation in favour of p . Let the corresponding elimination order be $e = (c_1, \dots, c_m = p)$, where c_i is the candidate eliminated in round i . Now, if the electorate were perfectly single-peaked, then from Lemma 3 we know that at each round only one of the rightmost or the leftmost candidate in the linear ordering L will be eliminated. But here we have exactly 1 maverick and so that claim is not always true. As a result, we have the following cases to consider.

Case 1: The first case is when at each round only one of the candidates at the ends in L is eliminated. This case is identical to the perfectly single-peaked case and from Lemma 4 we know that the reverse of the elimination order here will be single-peaked.

Case 2: Suppose there was some round i in which there was a candidate c_i eliminated such that c_i was neither the leftmost nor the rightmost candidate in the L remaining after $i - 1$ rounds. If $LV(c_j)$ denotes the number of last votes for c_j , then the fact that c_i was eliminated in the i th round implies that $LV(c_i) > LV(c_j), \forall j \neq i$. Now, since c_i was not the rightmost or leftmost candidate in L , the only way c_i can have last votes is if, in the i th round, the vote of the maverick had c_i at the end. But then, this also implies that from the $(i + 1)$ th round, the maverick is practically a dictator as for every round from i only the candidate placed last (for that round) in the maverick's vote will be eliminated. Therefore, in this case, the only way p can win is if it is placed at the top by the maverick. As a result, what we see here is that, if at all the manipulators' votes have any significance (it wouldn't matter how they vote if the first candidate to be eliminated in the election was a candidate not present at the either of the ends in L) in determining if p wins, that significance holds only for the first $i - 1$ rounds. Also, since the first $i - 1$ rounds only had one of the candidates at the ends in L eliminated, we know that there is at least one single-peaked order that has p at the top and $\{c_1, \dots, c_{i-1}\}$ in the last $i - 1$ places, where c_j is the candidate placed at the j th position from the bottom in the preference order.

In both the cases above, if the appropriate single-peaked order (which is the reverse of the elimination order in case 1, and a single-peaked order with p at the top and $\{c_1, \dots, c_{i-1}\}$ in the last $i - 1$ places for case 2) is determined, then all the manipulators can simply vote for that order and this in turn will result in a successful manipulation in favour of p . Doing this is enough because we know that in $\text{eliminate}(\text{veto})$ any elimination order that is achieved by an arbitrary set of manipulators' votes can be achieved if

all the manipulators vote the same way (by using the reverse of the elimination order) [11, Lemma 12]. \square

Proof of Theorem 36 contd. As a consequence of Lemma 5, the manipulators can now simply try out all possible single-peaked orders (that are consistent with L) which have p at the top. Since there are only a polynomial number of such orders (when the number of candidates are bounded), this can be done in polynomial time.

To prove the second part of the theorem, we can proceed the same way as in the second part of Theorem 32. Construct the following instance of CWCM with $a \ L \ b \ L \ p$ as the linear ordering and with S having a vote of weight $K + 2$ voting $(b > p > a)$, a vote of weight $K - 1$ voting $(b > a > p)$, a vote of weight 1 voting $(a > b > p)$, a vote of weight 1 voting $(p > a \sim b)$, a vote of weight 1 voting $(p > a > b)$, and a vote of weight 2 voting $(a > b \sim p)$. Note that, here, (p, a, b) is the maverick. In T , let each k_i have a vote of weight k_i . Use Partition for reduction. \square

4.3 Bribery

Faliszewski *et al.* were the first to look at the complexity of bribery in elections [22]. Subsequently, the problem was studied by Brandt *et al.* in single-peaked settings and there they showed that many of the combinatorial protections for bribery vanish when the preferences are restricted to being single-peaked [9]. Finally, Faliszewski *et al.* also studied the problem when the preferences are nearly single-peaked [24]. Here, we revisit the problem of bribery in single-peaked and nearly-single peaked settings and we try and see if bribery too, like manipulation, fits into our theme of reinstating combinatorial protections in single-peaked and nearly single-peaked elections through top-truncated voting.

4.3.1 Weighted-Bribery in Scoring Rules

Here, we first derive the results for 3-candidate scoring rules in single-peaked settings when only complete votes are allowed. Subsequently, we do the same when top-truncated ballots are allowed. The \mathcal{NP} -completeness proofs here use an idea that is similar to the one used by Faliszewski *et al.* in Theorem 4.9 [22], where they use a reduction from a modified version of the weighted manipulation problem to show that α -weighted-bribery is \mathcal{NP} -complete when it isn't the case that $\alpha_2 = \alpha_3 = \dots = \alpha_m$. Although it is possible to extend the results of the complete votes case here to any m -candidate scoring rule, thanks to

the complete characterization of weighted manipulation for scoring rules given by Brandt *et al.* [9], we stick to the 3-candidate case here because, to the best of our knowledge, such a complete characterization is not yet known when top-truncated ballots are allowed.

Let us first define the modified version of manipulation that we will use to reduce to the problem of weighted-bribery. The modified problem defined here is similar to the one used by Faliszewski *et al.* [22], with the only difference that in their problem all the manipulators need to have weights at least twice as much as the weight of the heaviest non-manipulator, while in our case we require that all the manipulators need to have weights at least thrice as much as the weight of the heaviest non-manipulator.

Definition 14 (CWCM'). *CWCM' is the same problem as CWCM with the restriction that each manipulative voter has a weight at least thrice as much as the weight of the heaviest non-manipulator.*

Theorem 37. *In single-peaked electorates, CWCM' with complete votes is \mathcal{NP} -complete for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) > 2(\alpha_2 - \alpha_3)$.*

Proof. To prove the above theorem for the complete votes case, we make use of Faliszewski *et al.*'s proof of Theorem 4.4 [27] which shows how CWCM is \mathcal{NP} -complete for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) > 2(\alpha_2 - \alpha_3)$. The first observation to make here is that the reduction in [27, Theorem 4.4] works even if we use Partition'. So, to prove the \mathcal{NP} -completeness of CWCM', the only thing we need to do is to make sure that the manipulators used in the reduction have weights at least thrice as that of the heaviest non-manipulator. This in turn can be ensured if we can somehow replace every non-manipulator who has a weight greater than the threshold (which here is one-third the weight of the lightest manipulator) with several non-manipulators each of whose weight is less than the threshold.

Again, in the proof of [27, Theorem 4.4], if we use Partition' for reduction then there will be $2n$ manipulators (instead of just 2 in the original proof) each of weight $3(2\alpha_1 - \alpha_2)K$ such that n of them vote for $(a > p > b)$ and the other n vote $(b > p > a)$. Additionally, each of manipulators will now have a weight of $3(\alpha_1 - 2\alpha_2)a_i$, where $\{a_1, \dots, a_n\}$ is an arbitrary instance of Partition' with $a_i \geq K, \forall i$ and $\sum_i a_i = 2nK$ (the rationale behind multiplying an additional factor of 3 to the weights of each manipulator and non-manipulator will be evident below). But now, since we need the weights of the manipulators to be at least thrice as much as the weight of the heaviest non-manipulator, we will have to replace each non-manipulator by at most

$$\left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{\left\lfloor \frac{1}{3}(3(\alpha_1 - 2\alpha_2))a_{\min} \right\rfloor} \right\rceil$$

non-manipulators, where $a_{\min} = \operatorname{argmin}_i \{a_i\}$. Since $a_{\min} \geq K$, and $(\alpha_1 - 2\alpha_2) \geq 1$, we have,

$$\left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{\left\lfloor \frac{1}{3}(3(\alpha_1 - 2\alpha_2))a_{\min} \right\rfloor} \right\rceil \leq \left\lceil \frac{3(2\alpha_1 - \alpha_2)K}{a_{\min}} \right\rceil \leq \lceil C \rceil$$

where $C = 3(2\alpha_1 - \alpha_2)$ is a constant. Now, since this splitting can be done in polynomial time, that completes the proof of the theorem. \square

We now show the result for weighted-bribery in scoring rules.

Theorem 38. *In single-peaked settings, weighted-bribery with complete votes is in \mathcal{P} for 3-candidate scoring rules when $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ and is \mathcal{NP} -complete otherwise.*

Proof. Let a, b , and p be the three candidates. Without loss of generality we can assume that $\alpha_3 = 0$. We will discuss both cases $(\alpha_1 \leq 2\alpha_2)$ and $(\alpha_1 > 2\alpha_2)$ separately.

Case 1: $\alpha_1 \leq 2\alpha_2$. Here we basically have two cases: i) when the input linear ordering of the candidates is $a \succ b \succ p$ and ii) when the input linear ordering of the candidates is $a \succ p \succ b$. In scoring rules, it is reasonable to assume that the bribed voters will always be made to vote in such a way that p will be at the top. Therefore, in the first case, since there is only one vote ($p > b > a$) that has p placed at the top, all the bribed voters will be bribed to vote ($p > b > a$). So now the only part remaining part here is to identify who to bribe. This can be done in the following way. Let us suppose that there was a successful bribery for a given instance. Since the only allowed votes for $a \succ b \succ p$ are $\{(a > b > p), (b > a > p), (b > p > a), (p > b > a)\}$, any successful bribery would have resulted in bribing x_1 voters initially voting ($a > b > p$), x_2 voters initially voting ($b > a > p$), and x_3 voters initially voting ($b > p > a$). Now, to identify these x_1, x_2 , and x_3 , we simply need to iterate through all the feasible (a solution (x_1, x_2, x_3) is feasible if the input set of votes has at least x_1 voters voting ($a > b > p$), at least x_2 voters voting ($b > a > p$), and at least x_3 voters voting ($b > p > a$)) integral solutions of $x_1 + x_2 + x_3 \leq k$, where k is the bribe limit, pick the x_1 heaviest voters voting ($a > b > p$), the x_2 heaviest voters voting ($b > a > p$), the x_3 heaviest voters voting ($b > p > a$), bribe all of them to vote ($p > b > a$), and check if it makes p a winner. If yes, we accept. Else, we continue on to the next set of solutions which satisfy the equation. Since there are only $O(k^3)$ such solutions, and since $k \leq n$, this can be done in polynomial time.

For the case $a \succ p \succ b$, the only allowed votes are $\{(a > p > b), (p > a > b), (p > b > a), (b, p, a)\}$, and so we can bribe a voter to vote either ($p > a > b$) or ($p > b > a$). Therefore, the main task here is to first identify the right voters to bribe and then decide on whether to

make them vote $(p > a > b)$ or $(p > b > a)$. To do this, let us assume there was a successful bribery in favour of p . This in turn would have resulted in bribing x_1 voters initially voting $(a > p > b)$, x_2 voters initially voting $(b > p > a)$, x_3 voters initially voting $(p > a > b)$, and x_4 voters initially voting $(p > b > a)$. Now, let $F = \{(x_1, x_2, x_3, x_4) \mid x_1 + x_2 + x_3 + x_4 \leq k \text{ and } (x_1, x_2, x_3, x_4) \text{ is feasible}\}$. For each $x = (x_1, x_2, x_3, x_4) \in F$, pick the heaviest x_1 voters voting $(a > p > b)$, the heaviest x_2 voters voting (b, p, a) , the heaviest x_3 voters voting $(p > a > b)$, and the heaviest x_4 voters voting (p, b, a) . Let S' be the set of all these voters. Next, calculate the scores of p, a , and b considering all the votes in $S - S'$, where S is the set of all votes given as part of the input. Let these scores be denoted by $s_{S-S'}(p)$, $s_{S-S'}(a)$, and $s_{S-S'}(b)$ respectively. Now for the case $aLpLb$, since $\alpha_1 \leq 2\alpha_2$, for any set of votes, p cannot lose to both a and b . Therefore the only possible cases for the scores $s_{S-S'}(p)$, $s_{S-S'}(a)$, and $s_{S-S'}(b)$ are the following. For each of them we mention how the bribery needs to be done.

1. $s_{S-S'}(p) \geq s_{S-S'}(a)$, $s_{S-S'}(p) \geq s_{S-S'}(b)$: In this case, for each voter in S' , make him or her vote either $(p > a > b)$ or $(p > b > a)$, if they aren't already voting for either of them.
2. $s_{S-S'}(b) > s_{S-S'}(p)$, $s_{S-S'}(a) \leq s_{S-S'}(p)$: In this case, for each voter in S' , make him or her vote $(p > a > b)$, if they aren't already.
3. $s_{S-S'}(a) > s_{S-S'}(p)$, $s_{S-S'}(b) \leq s_{S-S'}(p)$: Here, for each voter in S' , make him or her vote (p, b, a) , if they aren't already.

Doing the above takes only polynomial time since there are only $O(k^4)$ feasible solutions and $k \leq n$.

Case 2: $\alpha_1 > 2\alpha_2$. To show that weighted-bribery in this case is \mathcal{NP} -complete, we use a reduction from CWCM'. Note that to show this we use the linear ordering $a L p L b$ since there are polynomial time algorithms for both CWCM' and weighted-bribery when p is at one of the ends in the societal order. Now, given an arbitrary instance (C, S, T, p) of CWCM', we construct the following instance (C, V, p, k) where C is set of candidates, $V = S \cup T'$ where T' is the set of voters from T voting either $(a > p > b)$ or $(b > p > a)$. We set $k = |T|$.

It is easy to see that if a successful manipulation exists then bribery with at most $k = |T|$ bribes is possible. To prove the converse we basically need to show that any bribery with at most $|T|$ bribes is possible if we bribe only the voters from T' . For this, let us assume this were not the case and that there was a voter $v \in V - T'$ who was part

of the successful bribery in favour of p . By bribing v , the gain for p against a or b is at most $(\alpha_1 + (\alpha_1 - \alpha_2))w(v)$, where $w(v)$ is the weight of voter v . On the other hand if a voter $v' \in T'$ was bribed, then the gain for p against a or b is at least $(\alpha_1 - \alpha_2)w(v')$. But then, according to the restriction imposed on CWCM', any $v' \in T'$ has a weight at least thrice as much as the heaviest voter in $V' - T$. This in turn implies that $w(v') \geq 3w(v)$, and so we might as well bribe v' instead of v . As a result, any bribery in favour of p can be achieved by bribing only the voters from T' . Therefore, existence of a successful bribery implies that a successful manipulation exists in CWCM'. \square

For the case of top-truncated ballots, we proved in Theorem 28 that, in single-peaked settings, CWCM with top-truncated votes is \mathcal{NP} -complete for all 3-candidate scoring rules except plurality and veto when the evaluation scheme was round-down. Similarly, we also showed in Theorem 29 that CWCM with top-truncated votes is \mathcal{NP} -complete for all 3-candidate scoring rules except plurality when the evaluation scheme was average-score. Based on these two theorems it is easy to see that we can make similar ‘splitting’ arguments as in the proof of Theorem 37 to prove these results hold true even for CWCM'. As a result, we can state the following results which can be proved by using a reduction from CWCM' similar to the one shown in case 2 of Theorem 38.

Theorem 39. *For any 3-candidate scoring rule X that is not isomorphic to plurality or veto, in single-peaked electorates, weighted-bribery with top-truncated votes in X_{\downarrow} is \mathcal{NP} -complete.*

Theorem 40. *For any 3-candidate scoring rule X that is not isomorphic to plurality, in single-peaked electorates, weighted-bribery with top-truncated votes in X_{av} is \mathcal{NP} -complete.*

Note that we can similarly prove the corresponding results for nearly single-peaked electorates as well.

4.3.2 Weighted-Bribery in Eliminate(veto)

Here we look at the problem of weighted-bribery in eliminate(veto). First, we study the problem in single-peaked electorates and following that we look at the nearly single-peaked case. In both cases, yet again, we observe that allowing top-truncated voting increases the complexity of weighted-bribery from being in \mathcal{P} to being \mathcal{NP} -complete.

Theorem 41. *In single-peaked electorates, and in the unique winner model, for 3-candidate eliminate(veto),*

1. *weighted-bribery with **complete votes** is in \mathcal{P} .*
2. *weighted-bribery with **top-truncated votes** is \mathcal{NP} -complete.*

Proof. First we discuss the polynomial time algorithm for weighted-bribery when only complete votes are allowed.

Let a, b , and p be the three candidates. Without loss of generality, we need to consider only two linear orders: $a \prec b \prec p$ and $a \prec p \prec b$.

i) $a \prec b \prec p$: For the given linear ordering, only the following elimination orders are possible: $\{(a, b, p), (a, p, b), (p, a, b), (p, b, a)\}$, where (a, b, p) implies that a is eliminated in the first round, followed by b , and then p (which is the winner). We will consider each of these cases separately.

Case 1: (a, b, p) . Here no bribery is required as p is already the winner.

Case 2: (a, p, b) . In this case, it is easy to see that a simple greedy strategy would suffice. Since b cannot be eliminated in the first round (as there is no vote with b at the end according to L), we only need to make sure that p wins against b in the second round. This can be ensured by doing the following:

Starting with the heaviest voter among all voters who do not rank p above b , bribe him or her to vote $(p > b > a)$. If p is the winner after the bribe, then accept. If not, continue the same until number of voters bribed exceeds the limit k . If the limit exceeds, reject.

Case 3: (p, a, b) . Here for p to win, it has to be saved from elimination in the first round, and subsequently has to beat b in the second round. To do this, we can proceed in two phases: In the first phase, starting with the heaviest voter among all voters who do not rank a last, bribe them to vote $(p > b > a)$. After each bribe, check if $LV(a) > LV(p)$, where $LV(a)$ denotes the total weight of all the voters who have placed a last in their preference ordering. Once that is done, i.e. once $LV(a) > LV(p)$, we check if p is the winner. If yes, we accept. Else, we move on to the second phase where starting from the heaviest voter but now among all voters who do not rank p above b , we bribe them to vote $(p > b > a)$. Here again, after each bribe, we see if it results in p being the winner. If it does, we accept. Otherwise, if at any point during this algorithm the number of voters bribed exceeds the limit k , then we reject.

Case 4: (p, b, a) . Follow the exact same algorithm as in case 3.

ii) $a \prec p \prec b$: Here, only the following elimination orders are possible: $\{(a, b, p), (a, p, b), (b, a, p), (b, p, a)\}$. Again, we will consider each of these cases separately.

Case 1: $(a, b, p), (b, a, p)$. In both these cases no bribery is required as p is already the winner.

Case 2: (a, p, b) . Here the situation is a little more complicated since the voters can be bribed to either induce the elimination order (a, b, p) or (b, a, p) . But although this is the case, it is possible to prove (see Lemma 6) that inducing $(a > b > p)$ is enough as inducing the other would require bribing more voters. So now, this case is similar to case 2 for the linear order $a \succ b \succ p$ and we can use the algorithm described there.

Case 3: (b, p, a) . This is similar to case 2 above with a and b interchanged.

This concludes the proof of the first part.

To prove the second part, we show how Partition' can be used to prove that weighted-bribery is hard for eliminate(veto) when top-truncated votes are allowed. The proof essentially follows an idea similar to the one used for proving the corresponding result for manipulation in Theorem 32, only that we show this reduction through the modified partition problem. The main motivation behind using the modified partition problem is to basically facilitate a situation that will enable us to argue that the bribery can be restricted to a certain set of voters (who correspond to the manipulators in the corresponding manipulation instance).

Now, it is easy to see that the problem is clearly in \mathcal{NP} . To prove \mathcal{NP} -hardness, we show a reduction from an arbitrary instance $\{a_1, \dots, a_n\}$ of Partition', where $\sum_i a_i = 2nK$ and $a_i \geq K, \forall i$, to an instance of weighted-bribery (C, V, p, k) , where $C = \{a, b, p\}$ is the set of candidates, $a \succ b \succ p$ is the linear order of the candidates, and V is the set of the following voters.

1. For each a_i , construct a voter v_i whose weight is a_i and who votes for $(b > a > p)$. Let T be the set of all these voters.
2. Construct the following set of voters S : n voters of weight K each voting $(b > p > a)$, n voters of weight $K - 1$ each voting $(b > a > p)$, $n - 1$ voters of weight 1 each voting for $(b > a > p)$, 1 voter of weight 1 voting $(a > b > p)$, 1 voter of weight 2 voting $(b > p > a)$, 1 voter of weight 2 voting $(p > a \sim b)$, and 1 voter of weight 3 voting $(a > b \sim p)$.

Set the bribe limit $k = n$ and $V = S \cup T$.

First thing to notice is that, if a is eliminated in the first round then p cannot be a unique winner, no matter how we choose the n voters to bribe. Therefore the only way

to make p a unique winner is by making sure that a does not get eliminated. Now since the bribe limit is n , the maximum total weight of the voters bribed cannot exceed $2nK$. Also, using arguments similar to the one used in the proof of Theorem 32, we can see that any bribery with at most n bribes is possible only if there is an equal weight of nK that is assigned to both $(p > a \sim b)$ and $(a > b \sim p)$. Therefore, the only part that remains to be argued is to show that this bribery with at most n bribes can be achieved if we bribe only those voters in T .

Before we show this, let us see the objectives involved in the bribery of this instance. First is that any bribe results in changing the vote of the corresponding voter to either $(p > a \sim b)$ or $(a > b \sim p)$. Secondly, while bribing a voter, we always want the gain for p against every other candidate to be as much as possible. And lastly, we also need to ensure that while a is given enough score so that it is not eliminated in the first round, any bribe to vote $(a > b \sim p)$ does not result in the score of p decreasing.

Now, having seen the objectives, let us suppose that there was a voter $v \in S$ who was part of a successful bribery in favour of p . Before the bribe, v has one among $(b > p > a)$, $(b > a > p)$, $(a > b > p)$, $(p > a \sim b)$, and $(a > b \sim p)$ has his or her vote. But given the objectives, the only rational choice of a voter who has to be bribed is one who's current vote is $(b > a > p)$ (bribing one with $(a > b \sim p)$ is irrational since to compensate the reduction in the score of a we will have to bribe another voter; bribing a voter with $(b > p > a)$ is irrational since if this voter was bribed to vote $(p > a \sim b)$ then the gain for p would have been more if instead a voter voting $(b > a > p)$ would have been bribed, or otherwise if he or she was bribed to vote $(a > b \sim p)$ then another voter would have to be bribed to compensate the reduction in p 's score; bribing a voter with $(a > b > p)$ is irrational since bribing this voter has the same effect as bribing a voter with $(b > a > p)$ and the weights of all the voters voting $(b > a > p)$ is at least as much as the weight of the voter voting $(a > b > p)$). But then, all the voters in T are not only voting $(b > a > p)$ initially, but they also have weights which are at least as much as of those voters voting $(b > a > p)$ in S . Hence, any bribery that is possible with at most n bribes can be achieved if we bribe only the voters from T . Now, since the total weight of all voters in T is equal to $2nK$, this implies that p is a unique winner if and only if there is a partition. \square

Lemma 6. *For the linear order $a L p L b$, when the input elimination order is (a, p, b) , for a successful bribery it is enough to induce the elimination order (a, b, p) .*

Proof. Since the linear order is $a L p L b$, only the following votes are allowed: $\{(a > p > b), (p > a > b), (p > b > a), (b > p > a)\}$. Let us assume that the total weight of all the voters voting $(a > p > b)$ is W_1 , those voting $(p > a > b)$ is W_2 , those voting $(p > b > a)$ is

W_3 , and those voting $(b > p > a)$ is W_4 . Now, since the input elimination order is (a, p, b) , we have the following inequalities: (1) $W_1 + W_2 < W_3 + W_4$ (since a is getting eliminated in the first round, $LV(a) > LV(b)$, where $LV(a)$ is the total weight of all the voters who place a at the end), and (2) $W_1 + W_2 + W_3 < W_4$ (since p gets eliminated in the second round). A successful bribery for p can proceed in two ways. First is if the bribery induces the elimination order (a, b, p) and the other way is to induce the elimination order (b, a, p) . The task here is to prove that inducing (a, b, p) is better than inducing (b, a, p) , or in other words, prove that inducing (b, a, p) requires bribing at least one more voter than what is required to induce (a, b, p) .

To prove this, let us consider the case when the elimination order (a, b, p) is induced. Since the input already has a getting eliminated in the first round, we only need to look at the second round and see how much total weight should be bribed so that p is saved from elimination. Once a is eliminated, $LV(b) = W_1 + W_2 + W_3$, and $LV(p) = W_4$. Therefore, for p to win in the second round, the weight of votes we need to bribe is at least $D_{pb} = \left\lceil \frac{W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil$. Let this bribery require k_1 voters to be bribed. Moreover, note that all the bribed votes are of the form $(b > p > a)$ and they will be bribed to vote $(p > b > a)$.

Now, let us consider the second elimination order (b, a, p) . To induce this, that is to make b get eliminated in the first round, the total weight of votes we need to bribe is at least $D_{ab} = \left\lceil \frac{(W_3 + W_4) - (W_1 + W_2)}{2} \right\rceil = \left\lceil \frac{2W_3 + W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil = W_3 + \left\lceil \frac{W_4 - (W_1 + W_2 + W_3)}{2} \right\rceil = W_3 + D_{pb}$. As a result, among voters voting $(b > p > a)$ we need to bribe a weight of at least D_{pb} so as to get b eliminated in the first round. But then, this bribery requires at least k_1 voters (from above), which in turn implies that we need at least one more voter to be bribed here than in the case where the elimination order (a, b, p) was induced. Hence, when the input elimination order is (a, p, b) and the linear order $aLpLb$, for a successful bribery it is enough to induce the elimination order (a, b, p) . \square

Next, we look at the complexity of bribery for `eliminate(veto)` in 1-maverick single-peaked electorates.

Theorem 42. *In 1-maverick single-peaked electorates, in the unique winner model, for 3-candidate `eliminate(veto)`,*

1. *weighted-bribery with **complete votes** is in \mathcal{P} .*
2. *weighted-bribery with **top-truncated votes** is \mathcal{NP} -complete.*

Proof. To prove the first part, we proceed like in Theorem 41 and consider the two possible input linear orderings $a \succ b \succ L \succ p$ and $a \succ L \succ p \succ b$ separately, where a, b , and p are the three candidates.

i) $a \succ b \succ L \succ p$: Now as opposed to the case in single-peaked electorates, here b can be eliminated in the first round (due to the presence of the maverick) and so, the following elimination orders are possible: $\{(a, b, p), (b, a, p), (b, p, a), (a, p, b), (p, a, b), (p, b, a)\}$. We will consider each of these separately.

Case 1: $(a, b, p), (b, a, p)$. In both these cases, no bribery is required as p is already the winner.

Case 2: (b, p, a) . b can only be placed in the last place in his or her preference ordering by the maverick. Therefore, here, the fact that b is getting eliminated in the first round implies that the weight of the maverick is greater than the total weight of all voters who have placed a at the end and the total weight of all voters who have placed p at the end. This in turn implies that once b is eliminated, only the candidate above b in the maverick's preference order will get eliminated, or in other words the maverick in this scenario acts like a dictator. Hence, the only voter who needs to be bribed is the maverick.

Case 3: $(a, p, b), (p, a, b), (p, b, a)$. For all these cases, we can follow the exact same algorithm as outlined in the proof of Theorem 41 for the case of single-peaked electorates.

While we have covered all the cases for the linear order $a \succ b \succ L \succ p$, one question that could arise is: What if the input set of voters did not have a maverick? Is there anything more that can be achieved if we introduce a maverick while bribing? The answer to this is No. This is so because, consider the case if the input elimination order was (a, p, b) . Then the only rationale choice (among $(a > p > b)$ and $(p > a > b)$) of vote for a bribery-induced maverick according to L is $(p > a > b)$. Now, let us suppose that we were to introduce such a maverick through bribing. This in turn can result in two possibilities: i) b getting eliminated in the first round ii) a remains the one to be eliminated in the first round. In case i), the fact that b got eliminated in the first round as a result of introducing the maverick implies that there already existed a voter in the input whose weight was so high that bribing him alone would have anyway sufficed. And our greedy strategy would have anyway identified such a voter and bribed him or her to vote $(p > b > a)$. In the other case, the only objective this achieves is that it lowers the gap between the scores of p and b , which again would have been achieved by our greedy strategy as it always picks the heaviest voter who does not rank p above b and makes him or her vote $(p > b > a)$. Therefore there is nothing more we are achieving by introducing a maverick through bribery. Similarly, we can argue the same for the other elimination orders.

ii) $a \prec p \prec L \prec b$: For this linear order if the input elimination order were (p, a, b) or (p, b, a) then we can make the same argument as in the case 2 above to say that we need to bribe only the maverick and make him or her vote $(p > a > b)$ or $(p > b > a)$. For all the other cases we can proceed exactly as outlined in the proof of Theorem 41 for the case of single-peaked electorates. Additionally, the question of introducing a maverick through bribing does not arise here because both the maverick votes $(a > b > p)$ and $(b > a > p)$ are irrational choices as they both have p placed at the end.

This concludes the proof of the first part of theorem.

Next, we show the \mathcal{NP} -completeness result by using similar arguments as in the proof of second part of Theorem 41 with only a slight modification to the construction of T . Here, in T we have the following voters: n voters of weight K each voting $(b > p > a)$, n voters of weight $K - 1$ each voting $(b > a > p)$, $n - 1$ voters of weight 1 each voting $(b > a > p)$, 1 voter of weight 1 voting $(a > b > p)$, 1 voter of weight 1 voting $(p > a \sim b)$, 1 voter of weight 1 voting $(p > a > b)$, 1 voter of weight 2 voting $(b > a > p)$, and 1 voter of weight 2 voting $(a > b \sim p)$. Note that the voter voting $(p > a > b)$ is the maverick. \square

4.3.3 Is Weighted-Bribery for Weak-Condorcet Consistent Rules Always Easy?

Brandt *et al.* [9] showed that in single-peaked electorates weighted-bribery is in \mathcal{P} for all weak-Condorcet consistent voting rules (see [9, Theorem 4.4] for a more general result). The polynomial results in their theorem and the reason why it was possible to consider all weak-Condorcet consistent voting rules together was because of the well-known property of single-peaked electorates where it is guaranteed that there is always at least one weak-Condorcet winner (the top choices of the “median” voters are always weak-Condorcet winners). However, this property no longer holds when top-truncated votes are allowed. As has also been pointed out by Cantala [10], it is not even necessary that a weak-Condorcet winner exists in such settings. We illustrate this with the following example.

Example 4. Let $C = \{a, b, c, d, e\}$ with $a \prec L \prec b \prec L \prec c \prec L \prec d \prec L \prec e$ as the linear ordering. Let there be 5 voters and let their votes be $(a >_{v_1} b >_{v_1} c >_{v_1} d \sim_{v_1} e)$, $(b >_{v_2} c >_{v_2} d \sim_{v_2} e \sim_{v_2} a)$, $(c >_{v_3} d >_{v_3} e \sim_{v_3} b \sim_{v_3} a)$, $(d >_{v_4} e >_{v_4} a \sim_{v_4} b \sim_{v_4} c)$, and $(e >_{v_5} d >_{v_5} c \sim_{v_5} b \sim_{v_5} a)$, respectively. Now, it is easy to see that in the pairwise majority relation, a and b lose to d , c loses to b , and both d and e lose to c . Since everyone loses at least once, there is no weak-Condorcet winner.

Because of the above we can no longer consider all weak-Condorcet consistent rules together as done by Brandt *et al.* [9] and exploit the connection between the weak-Condorcet winner(s) and “median” voters to come up with polynomial time algorithms for weighted-bribery. In fact, next we show that for 3-candidate Baldwin’s rule (also known as Fishburn’s version of Nanson’s rule [40]), which is a weak-Condorcet consistent rule in single-peaked electorates [9], weighted-bribery is \mathcal{NP} -complete when top-truncated ballots are allowed. To show this we use an idea similar to the one used in Theorem 41.

Theorem 43. *In single-peaked electorates, weighted-bribery with **top-truncated votes** is \mathcal{NP} -complete for 3-candidate Baldwin’s rule.*

Proof. The problem is clearly in \mathcal{NP} . To prove \mathcal{NP} -hardness, we show a reduction from an arbitrary instance $\{a_1, \dots, a_n\}$ of Partition’, where $\sum_i a_i = 2nK$ and $a_i \geq K, \forall i$, to an instance of weighted-bribery (C, V, p, k) , where $C = \{a, b, p\}$ is the set of candidates, $a \succ b \succ p$ is the linear order of the candidates, and V is the set of the following voters.

1. For each a_i , construct a voter v_i whose weight is $2a_i$ and who votes for $(b \succ a \sim p)$. Let T be the set of all these voters.
2. Construct the following set of voters S : n voters of weight $2K$ each voting $(b \succ p \succ a)$, n voters of weight $2K$ each voting $(b \succ a \sim p)$, n voters of weight K each voting $(a \succ b \sim p)$, 2 voters of weight 1 each voting $(b \succ p \succ a)$, 2 voters of weight 1 each voting $(a \succ b \sim p)$, and 1 voter of weight 1 voting $(p \succ a \sim b)$.

Set the bribe limit $k = n$ and $V = S \cup T$.

First, observe that if a is eliminated in the first round then p cannot win, no matter how we choose the n voters to bribe. Therefore the only way to make p a winner is by making sure that a does not get eliminated. Now since the bribe limit is n , the maximum total weight of the voters bribed cannot exceed $4nK$. Also note that we need to bribe at least $4nK$ weight because otherwise either p or a will be eliminated in the first round. Additionally, it is possible to show that even now p can be a winner only if there is a total weight of $2nK$ voting $(p \succ a \succ b)$ and a total weight of $2nK$ voting $(a \succ p \succ b)$ (informally, this argument holds because failing to have equal votes on $(p \succ a \succ b)$ and (a, p, b) will cause either p or a to be eliminated in the first round). Therefore, the only part that remains to be argued is to show that any bribery with at most n bribes can be achieved if we only bribe the voters in T .

To prove this, let us assume the contrary and suppose that there was a voter $v \in S$ who was part of a successful bribery in favour of p . Before the bribe, v has one among

$(b > p > a)$, $(b > a \sim p)$, $(p > a \sim b)$ and $(a > b \sim p)$ has his or her vote. Now, given the fact that we always want the gain for p to be as much as possible, and because a needs to be given enough score to prevent it from being eliminated first, the only rational choice of voter who has to be bribed is one who's current vote is $(b > a \sim p)$ (bribing one with $(a > b \sim p)$ is irrational since to compensate the reduction in the score of a we will have to bribe another voter, bribing a voter with $(b > p > a)$ is irrational since if this voter was bribed to vote $(p > a > b)$ or $(a > p > b)$ then the gain for p would have been more if instead a voter voting $(b > a \sim p)$ was bribed). But then, all the voters in T are not only voting $(b > a \sim p)$ initially, but they also have weights which are at least as much as of those voters voting $(b > a \sim p)$ in S . Hence, any bribery that is possible with at most n bribes can be achieved if we bribe only the voters from T . Now, since the total weight of all voters in T is equal to $4nK$, this implies that p can be a winner if and only if there is a partition in Partition'. \square

Note that almost the same proof as above can be used to show that in single-peaked electorates CWCM with top-truncated ballots is \mathcal{NP} -complete for 3-candidate Baldwin's rule.

4.4 Is Top-truncated Voting Always Beneficial in SP Electorates?

Although we have seen instances, like in 3-candidate scoring rules with round-up evaluation scheme, where the complexity of manipulation decreases as a result of moving from a purely single-peaked setting to a setting where top-truncated votes are allowed, we haven't really seen examples of any other voting rule which shows this behaviour. Moreover, we also know that with a different evaluation scheme like round-down or average-score this behaviour is no longer seen for even 3-candidate scoring rules. Therefore, a natural question one could ask is: "What role does the evaluation scheme play? Is it possible that given a voting rule one can always construct an evaluation scheme so that it will be beneficial to allow top-truncated voting in single-peaked electorates?". Alternatively, one could also ask: "Is there a voting system for which it is always easy to manipulate when top orders are allowed?". We answer the former question in the negative and the latter one in the affirmative. We show that, as long as all the unranked candidates are assumed to be tied and are assumed to be ranked below the ranked candidates (which is the natural definition of a top-truncated vote), there is at least one voting system for which, irrespective of how

the top-truncated votes are dealt with, it is \mathcal{NP} -hard to manipulate in purely single-peaked settings, but is easy to manipulate when top-truncated votes are allowed.

Theorem 44. *There exists a voting system for which, in single-peaked settings,*

1. *CWCM with **complete votes** is \mathcal{NP} -complete.*
2. *CWCM with **top-truncated votes** is in \mathcal{P} .*

Proof. Let us first define the (artificial) voting system that we consider here.

Definition 15 (Artificial Voting Rule (AVR)). *Given the set, V , of voter preferences, the score of each candidate $c \in C$ is calculated as*

$$s(c) = \sum_{v \in V} \sum_{\substack{a \in C \\ a \neq c \\ c >_v a}} (m - \text{pos}(c))w(v)$$

where $\text{pos}(c)$ is the position of candidate c in the preference order of v if it is ranked ($\text{pos}(c) = i$ if c is ranked in the i th position by v) and is m otherwise, $m = |C|$, $w(v)$ is the weight of voter v , and $c >_v a$ denotes that c is ranked above a by v .

Example 5. Let $C = \{a, b, c, d, e\}$ be the set of candidates and let there be a single voter of weight w with the preference ordering ($d > b > a > e > c$). Then the scores of the candidates are $s(a) = 4w$, $s(b) = 9w$, $s(c) = 0$, $s(d) = 16w$, and $s(e) = w$.

So now we need to show that for AVR, in single-peaked settings, CWCM with complete votes is \mathcal{NP} -complete while CWCM with top-truncated votes is in \mathcal{P} .

We show the first part by a reduction from Partition. Given an arbitrary instance $\{k_i\}_{1 \leq i \leq t}$, $\sum_i k_i = 2K$, of Partition, construct the following instance of CWCM, where a, b , and p are the three candidates and $a L p L b$ is the linear ordering over the candidates. In S , let there be two voters of weight $7K$ each voting ($a > p > b$) and ($b > p > a$) respectively, and two voters of weight K each voting ($p > a > b$) and ($p > b > a$) respectively. In T , let each k_i have a vote of weight k_i . According to the rule defined above, the scores of a, b , and p are $29K, 29K$, and $22K$ respectively.

Suppose there exists a partition. Let those manipulators in one half vote ($p > a > b$) while those in the other half vote ($p > b > a$). As a result the scores of a, b , and p are all $30K$ and hence p is a winner.

Conversely, suppose there exists a manipulation in favour of p . In AVR it is reasonable to assume that all the manipulators place p first. So, now, let x and y be total weight of the

manipulators in T who vote $(p > a > b)$ and $(p > b > a)$ respectively. Since there exists a successful manipulation in favour of p , the score of p should be at least as much as that of a . Therefore, we have: $22K + 4x + 4y \geq 29K + x$. Using the fact that $x + y = 2K$, this simplifies to $x \leq K$. Doing the same with respect to p and b we have, $y \leq K$. But then since $x + y = 2K$, this implies that $x = K$ and $y = K$ and that there exists a partition.

For the second part, it is easy to see that the optimal strategy for the manipulators is to just vote $(p > a \sim b)$. \square

4.5 Conclusion

The central theme of this chapter was the reinstatement of combinatorial protections in single-peaked and nearly-single peaked electorates by allowing top-truncated voting. We observed this behaviour first in the case of manipulation and showed how for different voting protocols manipulation with complete votes was in \mathcal{P} whereas manipulation with top-truncated votes jumped to being \mathcal{NP} -complete. These results are summarized in Tables 4.1 and 4.2. Subsequently, we studied the problem of bribery where, again, similar behaviour was observed. These results are summarized in Table 4.3. In addition to the

Single-Peaked Electorates			
Voting Rule	CWCM (complete orders)	Voting Rule	CWCM (top-t orders)
3-X	\mathcal{P} , if $(\alpha_1 \leq 2\alpha_2)$ [27]	$3-X_{\downarrow}^1$	\mathcal{NP} -complete
	\mathcal{NP} -complete, if $(\alpha_1 > 2\alpha_2)$ [27]	$3-X_{av}^2$	\mathcal{NP} -complete
eliminate(Veto)	\mathcal{P}	eliminate(Veto)	\mathcal{NP} -complete
3-Copeland $^{\alpha}$ $\alpha \in [0, 1)$	\mathcal{P}	3-Copeland $^{\alpha}$, $\alpha \in [0, 1)$	\mathcal{NP} -complete

X : All scoring rules

X^1 : All scoring rules except plurality and veto

X^2 : All scoring rules except plurality

X^3 : All scoring rules except veto

Table 4.1: Change in complexity of CWCM when moving from complete votes to top-truncated votes in single-peaked electorates.

above results, we also showed an instance of a natural voting system ($\text{eliminate}(\text{veto})$) where, contrary to intuition, the complexity of manipulation, when top-truncated ballots are allowed, actually increases from being in \mathcal{P} in the general case to being \mathcal{NP} -complete in the single-peaked case. Finally, we concluded our discussion by showing the example of a voting system where allowing top-truncated voting isn't beneficial in the sense that it actually results in a decrease in the complexity of manipulation.

k -maverick Single-Peaked Electorates			
Voting Rule	CWCM (complete orders)	Voting Rule	CWCM (top-t orders)
$3\text{-}X^2$	\mathcal{NP} -complete [24]	$3\text{-}X_{\downarrow}^1$	\mathcal{NP} -complete
		$3\text{-}X_{av}^2$	\mathcal{NP} -complete
$\text{eliminate}(\text{Veto})$	\mathcal{P}	$\text{eliminate}(\text{Veto})$	\mathcal{NP} -complete

X : All scoring rules

X^1 : All scoring rules except plurality and veto

X^2 : All scoring rules except plurality

X^3 : All scoring rules except veto

Table 4.2: Change in complexity of CWCM when moving from complete votes to top-truncated votes in k -maverick single-peaked electorates.

Single-Peaked Electorates			
Voting Rule	CWCM (complete orders)	Voting Rule	CWCM (top-t orders)
$3\text{-}X$	\mathcal{P} , if $(\alpha_1 \leq 2\alpha_2)$	$3\text{-}X_{\downarrow}^1$	\mathcal{NP} -complete
	\mathcal{NP} -complete, if $(\alpha_1 > 2\alpha_2)$	$3\text{-}X_{av}^2$	\mathcal{NP} -complete
$\text{eliminate}(\text{Veto})$	\mathcal{P}	$\text{eliminate}(\text{Veto})$	\mathcal{NP} -complete
3-Baldwin's rule	\mathcal{P} [9]	3-Baldwin's rule	\mathcal{NP} -complete

X : All scoring rules

X^1 : All scoring rules except plurality and veto

X^2 : All scoring rules except plurality

X^3 : All scoring rules except veto

Table 4.3: Change in complexity of weighted-bribery when moving from complete votes to top-truncated votes in single-peaked electorates.

Chapter 5

Conclusions and Future Work

Collective decision making problems abound in human as well as multiagent contexts and typically voting is used as a mechanism to aggregate the preferences of the participating agents and to take a decision. However, voting is not without its problems as seminal results have shown that all reasonable voting rules are manipulable. As a result, there has been a lot of focus on circumventing such negative results and the goal of a recent body of literature has been to understand if and when computational complexity can be used as a barrier against different manipulative actions. While there have been many previous works which have studied the complexity of different manipulative actions, a disproportionate number of them have been under the assumption that the agents specify complete preference orderings over the set of candidates. Although it is an interesting setting to study, there are many practical situations where the agents may not be willing or may simply not be able to provide this information. It is these kind of scenarios that we study here, and in particular, we focus on situations where the agents are allowed to specify partial preferences in the form of top-truncated ballots. Therefore, although our work in this thesis belongs to this large domain of research, it is different in comparison to most of them since here we are focusing on a more practical setting where the agents are allowed to partially specify their preferences.

In looking at elections with top-truncated ballots, we addressed two related questions. First was the question of what the impact on complexity of manipulation is when top-truncated ballots are allowed. This was discussed in Chapter 3 and here we studied the problem of manipulation—both constructive and destructive—in weighted elections when the agents are allowed to specify top-truncated preferences and also looked at the impact on manipulation when there is uncertainty about the non-manipulators’ votes. We devoted most of this chapter studying the first problem—i.e. when the manipulators have

complete information on the non-manipulators—and in particular we provided general results for constructive and destructive manipulation in all scoring rules, elimination versions of all scoring rules, the plurality with runoff rule, a family of election systems known as Copeland^α, and the maximin protocol. The remainder of this chapter was devoted to exploring the second avenue i.e. when there is uncertainty about the non-manipulators’ votes. Here we discussed two possible ways in which the uncertainty can be modeled and we also showed that in both cases even individual manipulation under uncertainty was hard when constructive coalitional manipulation was hard. To the best of our knowledge, we are the first to study the impact on manipulation with top-truncated ballots when there is uncertainty about the non-manipulators’ votes.

The second issue we looked at is the question of what the impact of top-truncated ballots are on the complexity of manipulation and bribery when the electorate has structured preferences. In particular, we focused on single-peaked and nearly single-peaked preferences, and we observed a number of surprising results which in turn formed the theme of Chapter 4—of reinstating combinatorial protections in single-peaked and nearly-single peaked electorates by allowing top-truncated voting. We observed this behaviour first in the case of manipulation and showed how for different voting protocols manipulation with complete votes was in \mathcal{P} whereas manipulation with top-truncated votes jumped to being \mathcal{NP} -complete. These results were followed by the results for bribery where, again, we observed similar behaviour for the voting rules considered. In studying the above two, we note that, to the best of our knowledge, we are the first to systematically look at the impact on complexity of manipulative actions when the electorate is single-peaked or nearly single-peaked and when top-truncated preferences are allowed. In addition to the above results, we also showed an instance of a natural voting system (`eliminate(veto)`) where, contrary to intuition, the complexity of manipulation, when top-truncated ballots are allowed, actually increases from being in \mathcal{P} in the general case to being \mathcal{NP} -complete in the single-peaked case. Finally, we concluded our discussion in Chapter 4 by showing the example of a voting system where allowing top-truncated voting isn’t beneficial in the sense that it actually results in a decrease in the complexity of manipulation.

There are numerous possible avenues for future work. Foremost would be to consider other types of partial preferences like bottom orders, weak orders etc. Although Fitzsimmons and Hemaspaandra have done some work in this direction [29], they only look at specific protocols and moreover only consider the general setting—i.e. when the preferences of the agents do not have a particular structure. Especially in the case of structured preferences, it would be interesting to see if similar observations as in Chapter 4 can be made. Second, while studying the impact of top-truncated ballots on the complexity of manipulative actions in structured electorates, we have only considered the problems of

manipulation and bribery, but not control. Therefore, we feel that it would be worthwhile to see if similar observations can be made for the problem of control as well. Third, while considering nearly single-peaked preferences in this thesis, we have talked about only one notion of nearness, namely the k -maverick notion. However, there are several other notions of nearness (see the work of Erdélyi *et al.* [20]) and seeing if we can obtain similar results for them as well would be interesting. Fourth, we have considered only weighted elections in this work, but we believe that looking at the unweighted case would be even more insightful as it would help us understand the inherent “hardness” that is embedded in the voting rule itself. Often what seems to be happening in weighted elections is that the computational complexity seems to be arising mainly from the weights themselves, thus denying us the opportunity to understand the voting rule better. Therefore, it is definitely something to be considered as a future research direction. With regards to the impact of partial voting on the complexity of manipulative actions, to the best of our knowledge, there is only one work by Narodytska and Walsh that has some results on unweighted elections [38]. Finally, a possible criticism regarding the results in this work (and in fact any work which considers the same notions) could be that the results are in the worst-case and that we use \mathcal{NP} -hardness—which in many cases does not necessarily reflect the actual difficulty in practice—as the complexity measure. Therefore, one long term research direction would be to look at the average-case complexity for manipulative actions in elections. There has been some work that has been done in this direction (for e.g. see the work of Friedgut *et al.* [30]), but we believe that there is still scope for much more.

References

- [1] Barberá, S.: Indifferences and domain restrictions. *Analyse & Kritik* 29(2), 146–162 (2007)
- [2] Bartholdi, J., Trick, M.A.: Stable matching with preferences derived from a psychological model. *Operations Research Letters* 5(4), 165–169 (1986)
- [3] Bartholdi, J.J., Tovey, C.A., Trick, M.A.: How hard is it to control an election? *Mathematical and Computer Modelling* 16(8-9), 27–40 (1992)
- [4] Bartholdi III, J.J., Orlin, J.B.: Single transferable vote resists strategic voting. *Social Choice and Welfare* 8(4), 341–354 (1991)
- [5] Bartholdi III, J.J., Tovey, C.A., Trick, M.A.: The computational difficulty of manipulating an election. *Social Choice and Welfare* 6(3), 227–241 (1989)
- [6] Baumeister, D., Faliszewski, P., Lang, J., Rothe, J.: Campaigns for lazy voters: Truncated ballots. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. pp. 577–584 (2012)
- [7] Black, D.: On the rationale of group decision-making. *The Journal of Political Economy* pp. 23–34 (1948)
- [8] Black, D., Newing, R.A., McLean, I., McMillan, A., Monroe, B.L.: *The theory of committees and elections*. Springer (1958)
- [9] Brandt, F., Brill, M., Hemaspaandra, E., Hemaspaandra, L.A.: Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research* pp. 439–496 (2015)
- [10] Cantala, D.: Choosing the level of a public good when agents have an outside option. *Social Choice and Welfare* 22(3), 491–514 (2004)

- [11] Coleman, T., Teague, V.: On the complexity of manipulating elections. In: Proceedings of the 13th Australasian Symposium on Theory of Computing. pp. 25–33 (2007)
- [12] Conitzer, V.: Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research* 35, 161–191 (2009)
- [13] Conitzer, V., Sandholm, T., Lang, J.: When are elections with few candidates hard to manipulate? *Journal of the ACM (JACM)* 54(3), 14 (2007)
- [14] Copeland, A.H.: A reasonable social welfare function. In: Seminar on applications of mathematics to social sciences, University of Michigan (1951)
- [15] Doignon, J.P., Falmagne, J.C.: A polynomial time algorithm for unidimensional unfolding representations. *Journal of Algorithms* 16(2), 218–233 (1994)
- [16] Duggan, J., Schwartz, T.: Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare* 17(1), 85–93 (2000)
- [17] Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International Conference on World Wide Web. pp. 613–622 (2001)
- [18] Emerson, P.: The original Borda count and partial voting. *Social Choice and Welfare* 40(2), 353–358 (2013)
- [19] Ephrati, E., Rosenschein, J.S.: Multi-agent planning as a dynamic search for social consensus. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI). vol. 93, pp. 423–429 (1993)
- [20] Erdélyi, G., Lackner, M., Pfandler, A.: Computational aspects of nearly single-peaked electorates. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI) (2013)
- [21] Escoffier, B., Lang, J., Öztürk, M.: Single-peaked consistency and its complexity. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI). pp. 366–370 (2008)
- [22] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: How hard is bribery in elections? *Journal of Artificial Intelligence Research* pp. 485–532 (2009)

- [23] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: Using complexity to protect elections. *Communications of the ACM* 53(11), 74–82 (2010)
- [24] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence* 207, 69–99 (2014)
- [25] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research* pp. 275–341 (2009)
- [26] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: A richer understanding of the complexity of election systems. In: *Fundamental Problems in Computing*, pp. 375–406. Springer (2009)
- [27] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation* 209(2), 89–107 (2011)
- [28] Fishburn, P.C.: Condorcet social choice functions. *SIAM Journal on applied Mathematics* 33(3), 469–489 (1977)
- [29] Fitzsimmons, Z., Hemaspaandra, E.: Complexity of manipulative actions when voting with ties. In: *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT)*. pp. 103–119 (2015)
- [30] Friedgut, E., Kalai, G., Nisan, N.: Elections can be manipulated often. In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. pp. 243–249. IEEE (2008)
- [31] Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA (1979)
- [32] Gibbard, A.: Manipulation of voting schemes: A general result. *Econometrica: journal of the Econometric Society* pp. 587–601 (1973)
- [33] Hemaspaandra, E., Hemaspaandra, L.A.: Dichotomy for voting systems. *Journal of Computer and System Sciences* 73(1), 73–83 (2007)
- [34] Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence* 171(5), 255–285 (2007)

- [35] Konczak, K., Lang, J.: Voting procedures with incomplete preferences. In: Proceedings of the IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling. vol. 20 (2005)
- [36] Lackner, M.: Incomplete preferences in single-peaked electorates. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI). pp. 742–748 (2014)
- [37] Lu, T., Boutilier, C.: Multi-winner social choice with incomplete preferences. In: Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence. pp. 263–270. AAAI Press (2013)
- [38] Narodytska, N., Walsh, T.: The computational impact of partial votes on strategic voting. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI). pp. 657–662 (2014)
- [39] Niemi, R.G., Wright, J.R.: Voting cycles and the structure of individual preferences. *Social Choice and Welfare* 4(3), 173–183 (1987)
- [40] Niou, E.M.: A note on Nanson’s rule. *Public Choice* 54(2), 191–193 (1987)
- [41] Pennock, D.M., Horvitz, E., Giles, C.L.: Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI). pp. 729–734 (2000)
- [42] Satterthwaite, M.A.: Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10(2), 187–217 (1975)
- [43] Walsh, T.: Uncertainty in preference elicitation and aggregation. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI). vol. 7, pp. 3–8 (2007)
- [44] Xia, L., Conitzer, V.: Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research* 41(2), 25–67 (2011)